

AD-A035 454

WHOI-77-2  
GRAVITY DATA PROCESSING PROGRAMS.(U)

FEB 77

CARL BOWIN

UNCLASSIFIED WOODS HOLE OCEANOGRAPHIC INSTITUTION, MA

1 OF 6  
AD-A  
035 454



U.S. DEPARTMENT OF COMMERCE  
National Technical Information Service

AD-A035 454

GRAVITY DATA PROCESSING PROGRAMS

WOODS HOLE OCEANOGRAPHIC INSTITUTION  
WOODS HOLE, MASSACHUSETTS

FEBRUARY 1977

# **N O T I C E**

**THIS DOCUMENT HAS BEEN REPRODUCED FROM THE  
BEST COPY FURNISHED US BY THE SPONSORING  
AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CER-  
TAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RE-  
LEASED IN THE INTEREST OF MAKING AVAILABLE  
AS MUCH INFORMATION AS POSSIBLE.**

ADA035454

AD NO. 7  
DDC FILE COPY

WHOI-77-2

GRAVITY DATA PROCESSING PROGRAMS

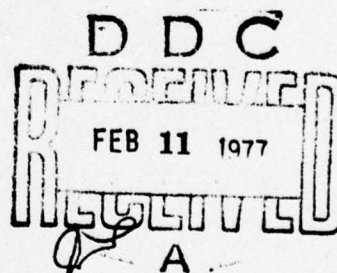
By

CARL BOWIN

WOODS HOLE OCEANOGRAPHIC INSTITUTION  
Woods Hole, Massachusetts 02543

January 1977

TECHNICAL REPORT



*Prepared for the Office of Naval Research under  
Contract N00014-74-C0262 NR 083-004.*

*Reproduction in whole or in part is permitted  
for any purpose of the United States Government.  
In citing this report in a bibliography, the  
reference should be followed by the phrase:  
UNPUBLISHED MANUSCRIPT.*

*Approved for public release; distribution  
unlimited.*

Approved for Distribution:

*John I. Ewing*  
John I. Ewing, Chairman  
Department of Geology & Geophysics

# ABSTRACT

A summary and documentation of a family of computer programs that have been developed by the gravity group at the Woods Hole Oceanographic Institution is presented. The programs provide for format conversion, computation of the regional gravity field from spherical harmonic coefficients, selective data retrieval, graphic display, and construction of two- and three-dimensional structure models and the computation of the gravitational attraction of those models.

## GRAVITY DATA PROCESSING PROGRAMS

## INTRODUCTION

This report is a summary and documentation of a family of computer programs that have been developed by the gravity group at the Woods Hole Oceanographic Institution. The programs documented here provide for format conversion, computation of the regional gravity field from spherical harmonic coefficients, selective data retrieval, graphic display, and construction of two- and three-dimensional structure models and the computation of the gravitational attraction of those models.

Many of the programs in this report have been used and modified for more than ten years. During this time six substantially different computer systems have been available to us. These are an Autonetics Recomp II, General Electric 225, IBM 7090 and 7094, IBM 1710 (shipboard), XDS Sigma-7, and Hewlett-Packard 2114, 2116, and 2100 (shipboard). Thus the programs have evolved not only because of changing needs and experience, but also because of different system hardware and software constraints. Artifacts reflecting this evolution are evident in some of the programs.

In the interpretation of gravity data and the creation of structure models of earth features, other geophysical and geological information is important. Gravity potential information alone does not define a unique mass distribution, and hence additional information is required to limit the possibilities. Therefore, we have added the capability for retrieving and displaying other types of data which are available in digital form. At the present time these data types include seismicity, seismic refraction profiles, and location of active volcanoes. The seismic refraction profiles are derived from a compilation of crustal seismic refraction profiles prepared by McConnel Jr. and McTaggart-Cowan of the University of Toronto in 1963 and from five supplements (No. 1 by Gupta and McTaggart-Cowan, 1964; No. 2 by Gertner, 1967; No. 3 by Gertner and Farquhar, 1968; No. 4 by Gertner and Farquhar, 1971; and No. 5 by Gertner and Farquhar, 1972). Supplements numbers 2 through 5 were sponsored by the Federation of Astronomical and Geophysical Services of I.C.S.U. Subsequent to the fifth supplement, financial assistance to the University of Toronto by U.N.E.S.C.O. for this compilation terminated, and unfortunately, this compilation effort has ceased. For our utilization of the seismic refraction compilation,

we find a single record per refraction line a more convenient format than the one- or two-record format prepared by the University of Toronto. We prefer magnetic tape or disc for data storage and accordingly are not limited by the 80 character record length of punched cards. We have incorporated additional data as we have had particular needs.

The file of locations of active volcanoes was originally coded from the Catalogue of Active Volcanoes. Volcanoes on New Zealand were added from Thompson (1964), and those in Alaska and the Aleutian Arc were added from Foster et al. (1966) and Coats (1950). More recently, IAVCEI has prepared data sheets of the post-Miocene volcanoes of the world (IAVCEI, 1975). A deck of cards based on these data sheets was obtained in December 1975 from NGSDC.

Additional data types can be incorporated into our programs relatively easily. Location and certain other characteristics of Deep Sea Drilling Program (DSDP) drill hole sites is a file of interest. We hope that a source for a global compilation of seismic slip mechanisms might be found.

#### PROGRAMS

A diagram outlining the functions served by the programs documented in this report is given as Figure 1. This diagram

serves as an index to the utilization of the family of programs, and it is intended to be largely self-explanatory. Table 1 lists the programs documented here and provides a summary statement of the purpose of each program. Table 1, together with figure 1, enables the reader to quickly find programs to meet his need.

Documentation for the programs themselves follows the references cited section. The programs are ordered alphabetically, and for each program the characteristics and operational parameters are described first, followed by a section containing listings of the source coding. Subroutines required by these programs are then given alphabetically in the section after the program listings. Standard system routines and those of a normal Fortran subroutine library are not reproduced. Normally, only a source listing is given for each subroutine.

## ACKNOWLEDGMENTS

Documentation of programs is a tedious activity, which normally seems to be deferred, awaiting a less busy time. The less busy time is an elusive quantity that never seems to arrive, and in the interim more programs are written and old programs modified to meet new requirements. After awhile, the original programmer or the modifier often has moved on to other pastures and is no longer available to help with documentation. Although I firmly believe in the importance of documenting programs when they are written, I personally have not been very faithful to that belief. Thus the existence of this report is largely due to the efforts of others. I particularly want to thank Allin Folinsbee and Leon Gove for being far more rigorous than I in documenting their programming efforts, Julie Milligan for a major contribution in the early stages of preparation of this report, Carolyn Dean for her efforts in the later stages, and Nan Galbraith, Leon Gove and Christine Wooding for their help in its completion. Allin is now at the Bedford Institute of Oceanography, Halifax, Nova Scotia; Julie is at the University of Auckland in Auckland, New Zealand and Carolyn is teaching high school in Falmouth, Massachusetts.

The preparation of this report was supported by the Office of Naval Research under contract N00014-74-C-0262; NR083-004.

Table 1

## PROGRAM DICTIONARY

ABSTGC	- ABSTRACTS GCON data at a spacing of 10 nautical miles.
CHART	- Plots data on Mercator charts
CHARTG	- Plots GSUM data on Mercator charts. Much faster than CHART for GSUM data.
CONV67	- Converts gravity data at Potsdam system to IGSN71 and International Gravity Formula 1967.
CR2G	- Converts land gravity meter counter readings to observed gravity.
CR2G67	- Like CR2G, but calculates anomalies referenced to International Gravity Formula 1967.
CRWT3	- Calculates the pressure at the base of a crustal column ( $\text{Kg}/\text{cm}^2$ ).
DMABLK	- Converts blocked DMA format data to blocked GSUM format data.
DMAF	- Converts digitized position in inches to latitude and longitude.
DMOD	- Punches digitized polygon points for crustal models digitized on a digitizing table.
GFLD1	- Calculates regional free-air gravity anomalies for a given region from spherical harmonic coefficients.
GFLD2	- Calculates regional free-air gravity anomalies from spherical harmonic coefficients entered at run time.
GRAFG2	- Plots one variable versus another.
GRAV1	- Converts data input at format of 8 July 1969 to GSUM format.

Table 1 (continued)

GSTOG	- Converts data input in SEAG1 or SEAG2 format to 128-character GSUM records.
GSTOG67	- Like GSTOG, but checks input for 1930 or 1967 gravity formula reference. Output referenced to 1967 International Gravity Formula.
G3DCP	- Computes gravity anomaly (for both flat and curved planetary surfaces), potential field (for flat surface), and mass per unit area for a set of polygonal laminae comprising a three-dimensional crustal structure model.
G3DCPREP	- Combines G3DCP input bodies into 1 file for processing.
HIG	- Converts gravity data in the format used by the Hawaii Institute of Geophysics (HIG) to GSUM format.
LSORT	- Checks laminae of G3DCP format for minimum thickness and counts them.
MODPLOT	- Plots data for preparation of structure models of the earth's crust and plots the output tapes from TALPLOT16.
NOAA	- Converts gravity data in NOAA format to WHOI GSUM format.
PROFG	- Profiles GSUM data
PROJ4	- Projects data onto a given line providing the data is within a given area and within a given distance from the line.
RETRIEVE	- System processor for retrieving data from data files.
SAINT2	- Interpolates data at even intervals.
SELSP	- Selects data output by the CRWT3 program on the basis of a given parameter.
SPFMT	- Converts seismic refraction column data in University of Toronto World Seismic Refraction Profile Compilation format to WHOI SPFMT format.
TALPLOT16	- Computes gravity anomalies and mass per unit area for a set of two-dimensional polygons.

TABLE 2

## SUBROUTINES REQUIRED BY THE PRECEDING PROGRAMS:

ALDT	GETX
ANOV2	GETY
ANOV3	GINOT
AREAK	GINTF
CALSC	GI67F
CDATE	GRIDG
CHGMT	GRID2
COORR	INCEP
DISAZ	ISW(1)
DMTOR	M2DY
DNAV	NAVIN
DREC	NAVOT
DY2M	OBG
ENDLT	BLINE
EVIL	PINOT
EXDT	PLANET
FIND	PLOTA
FLD2	RETBY
GETC	RTDM2
GETF	RTODM
GETG	SIMUL
GETGA	SPLOT
GETGC	SPOT
GETGS	SPOT2
GETH	TIDAL
GETL	VETBY
GETM	WEIG2
GETP	YBLIKI
GETS	YBLIKO
GETST	YINOT
GETV	

8a  
FIGURE 1

DATA TYPE	GRAVITY				SEISMICITY	SEISMIC REFRACTION PROFILES	ACTIVE VOLCANOES
SOURCE	WHOI Ship's SEAG1 fmt SEAG2 fmt	WHOI land data 8 July 1969	other land and marine data various fmts	Regional Gravity field Spherical Harmonic Coefficients		University of Toronto World Compilation	Catalogue of Active Volcanoes IAVCEI
Conversion Programs	GSTOG GSTOG67 CONV67	CR2G CR2G67 CONV67	GRAV1 DMABLK HIG NOAA DMAP	GFLD1 GFLD2	ISORT	SPFMT	
Master Data File	Gravity Data Library (EDL) GSUM format See Monget and Bowin (1974) for description and discussion of organizational philosophy				World seismicity Data Library geographically sorted	World Seismic Refraction Library at SPFMT format	World Active Volcanoe Library
Selective data retrieval, Graphic, display, Three-dimensional structure model Programs	→ CHARTG → CHART → GRAFG2 → PROFG → G3DCP			→ CHARTG → CHART	→ RETRIEVE → CHART	→ CHART → CRWT3 → SELSP	→ CHART
Data Projection	PROJ4						
Two-dimensional structure modeling				→ SAINT2			
				→ MODPLCT			
				→ DMOD			
				→ TALPLOT16			→ MOD

## REFERENCES CITED

Coats, R. R., Volcanic Activity in the Aleutian Arc, Geol. Surv. Bull., 974-B, 35-47, 1950.

Foster, H. L., R. B. Forbes, and D. M. Ragan, Granulite and Peridotite Inclusions from Prindle Volcano, Yukon-Tanana Upland, Alaska, U. S. Geol. Surv. Prof. Paper 550-B, p. B115-B119, 1966.

Gertner, B., Crustal Seismic Refraction Profiles - A Compilation (Supplement No. 2), Federation of Astronomical and Geophysical Services of I.C.S.U., University of Toronto, 9 p., 1967.

Gertner, B., and R. M. Farquhar, Crustal Seismic Refraction Profiles, A Compilation (Supplement No. 3), Federation of Astronomical and Geophysical Services of I.C.S.U., University of Toronto, 20, 1968.

Gertner, B., and R. M. Farquhar, Crustal Seismic Refraction Profiles- A Compilation (Supplement No. 4), Federation of Astronomical and Geophysical Services of I.C.S.U., University of Toronto, 23, 1971.

Gertner, B., and R. M. Farquhar, Crustal Seismic Refraction Profiles- A Compilation (Supplement No. 5), Federation of Astronomical and Geophysical Services of I.C.S.U., University of Toronto, 20, 1972.

Gupta, R. A., and G. H. McTaggart-Cowan, Crustal Seismic  
Refraction Profiles - A Compilation (Supplement No. 1)  
University of Toronto 13, 1964.

I.A.V.C.E.I. Working Group, Data sheets of the Post Miocene  
Volcanoes of the World with Index Maps, 61, 1973.

International Association of Volcanology (Ed.), Catalogue of the  
Active Volcanoes of the World Including Solfatara Fields,  
part 1-21, 1951-1967.

McConnel, R. K., Jr., and G. H. McTaggart-Cowan, Crustal Seismic  
Refraction Profiles: A Compilation, University of Toronto,  
171 p, 1963.

Monget, J.-M., and C.O. Bowin, A Gravity Data Library: Organization  
and Effective Utilization, Technical Rept. W.H.O.I. 74-33,  
43p., 1974.

Thompson, B.N., Quaternary Volcanism of the Central Volcanic  
Region, New Zealand Jour. Geol. and Geophys., Vol. 7,  
No. 1, p. 45-66, 1964.

NAME: ABSTGC

TYPE: Main Program

PURPOSE: To abstract GCON data at a spacing of 10 nautical miles

MACHINE: XDS Sigma 7

PROGRAM CATEGORY: Statistical

DESCRIPTION:

The program processes one degree of GCON data at a time. The data is decoded and if the data falls within the degree square of concern the free air anomaly and height is added to the appropriate 10 nautical mile square value and the position is checked to see if this point is the closest to the center of the square. If it is the closest, the values and the position are retained. When all the data for the degree square is processed the GABS data record is written to the output device.

INPUT:

PARAMETER CARDS (via F:105)

- 1) Sense switch card
  - SSW(46) = 0 no effect
  - SSW(46) = 2 process within bounds and use D.L.T.
- 2) Geographic Bounds (in degrees)
  - Top (KDTOP, I5)
  - Bottom (KDBOT, I5)
  - Left (KDLFT, I5)
  - Right (KDRGT, I5)
- 3) D.L.T. deck if applicable

GCON DATA (via F:1)

Data in GCON format blocked 22 x 50.

OUTPUT:

GABS DATA (via F:2)

Data in GABS format. Data in each physical record is all the data for one degree square. The first logical record of each physical record is the whole-part of the latitude and longitude. Following are 36 logical records, one each for the 10 nautical mile square (see figures 1 and 2)

## ABSTGC (continued) page 2

USAGE:SAMPLE RUN

```

!JOB
!LIMIT (9T,2)(CORE,20),(TIME,XY)
!MESSAGE I/P tape info
!MESSAGE O/P tape info
!ASSIGN F:1(DEVICE,9T),(SN,XXXX), (IN),(TRIES,10)
!ASSIGN F:2(DEVICE,9T),(SN,YYYY),(OUT),(TRIES,10)
!LOAD (BI),(UNSAT,(312),(3))
      ABSTGC object deck

```

```

!RUN
!DATA

```

Parameter cards

!EOD

RESTRICTIONS:

If bounds are to be checked a D.L.T. deck must be provided

STORAGE: 16K words

SUBPROGRAMS REQUIRED: ISW, FORTRANIV Library

TIMING: Thru-put time is about 3000 logical records/minute

PROGRAMMER: Lee Gove

ORIGINATOR: Carl Bowin

DATE: 15 October 1975

NAME: CHART

TYPE: Program

PURPOSE: Plot data on Mercator charts

SOURCE LANGUAGE: Sigma-7 Fortran 4

MACHINE: Sigma-7

PROGRAM CATEGORY: Graphical Display

DESCRIPTION:

Plots Mercator chart at specified scale, draws track and annotates with specified parameter.

INPUT:

Input formats: FIXSE, SEAG1, GSUM, MBATR, CALCM, and tabulations of refraction, earthquake, volcano, heat flow data. There is also a user specified format.

Program plots a 1/2-inch fiducial square in lower right corner of chart. If sides of square are offset means pen hit stops or lost registration in course of plotting.

1st card

(20A4) Label- up to 80 characters, plotted vertically on left-hand margin of chart.

2nd card Sense switch options

ISW(0) - ISW(79) (8011) Put Sense switch (0) option in column 80, all others in column corresponding to switch number.

Optional card

If ISW(10) = 1 on card 2, put four-character name of input tape here, format (A4). Using this option (subroutine MOUNT) it is possible to generate a plot tape with several plots per job separated by EOF, from one or more input tapes. This card is never used when input data is in GSUM format. For MOUNT cards for GSUM format, see card(s) seven below.

3rd card (2(3I2,I4,5X), 3I5)

Column

1,2	ISTDA	Start date for processing, for example
3,4	ISTMO	0204720341 means 2 February 1972 0341Z
5,6	ISTYR	If blank, plotting begins with first
7,10	ISTHM	record.

## 3rd card continued

## Column

16,17 IENDA

18,19 IENMO

20,21 IENYR

22-25 IENHM

End date for processing - if blank,  
plotting will continue until EOD or  
EOF is encountered

31-35 ISKP

No. of records to be skipped at start  
of job, is much faster than using start  
date only.

36-40 ISFIL

41-45 IBCKUP

No. of files to be skipped at start of job.  
For making a series of overlapping plots  
from the same file. Number of points  
common to this plot and the next. Tape  
is backed up IBCKUP+1 records before  
restarting program.

## 4th card (F10.0,215,1X,A4,415)

## Column-

1-10 SINCH = Inches/one degree of longitude in floating point

15 ITRK = 0 for deleting track between plotting points  
1 for including track

20 LCNT = N, for plotting every N'th point only

25 NDEG = for plotting every NDEG intermediate grid  
line for integer degree bounds and every  
NDEG minutes for non-integer degree bounds.27-30 NUMPL = plot number to be annotated in lower left-  
hand corner on plot (A4)

35 NPTA = N, to annotate every N'th plotted point

39,40 JFMT = Data format code

= 1, FIXSE Format

= 2, SEAG1 Format

= 3, GSUM Format

= 4, MBATR Format

= 5, CALCM Format

= 6, STATN Format

= 7, SPFMT Format

= 8, World seismicity Format

= 9, Active Volcanoes

= 10 Heat Flow

= 11 Lunar Data

= 12 User supplied

44,45 NX = tells what value is to be annotated beside  
point - the value to be annotated is a  
function of NX and JFMT. For all formats  
NX = 0 omits annotation beside data point50 NFILE = No. of files on the input tape to be  
plotted on the same grid

= 0 will still plot first file

JFMT = 1, FIXSE Format - navigation

- NX = 1 for time, and date at change of date
- = 2 for month
- = 3 for year
- = 4 for day
- = 5 for zone

JFMT = 2, SEAG1 Format - gravity

- NX = 1 for time, and date at change of day
- = 2 for water depth in corrected meters
- = 3 for free air anomaly
- = 4 for Bouguer anomaly
- = 5 for speed in knots
- = 6 for heading in degrees
- = 7 for Eotvos correction
- = 8 for Matthews Table number
- = 9 for low order 3 digits of total magnetic field intensity
- = 10 for total regional magnetic field (not implemented)
- = 11 for residual magnetic value (not implemented)
- = 12 for negative speed
- = 13 for negative water depth
- = 14 for negative Eotvos correction
- = 15 for negative Free Air anomaly
- = 16 for negative heading
- = 17 for total magnetic field intensity
- = 18 for uncorrected depth in meters
- = 19 for uncorrected depth in fathoms

JFMT = 3, GSUM Format, - gravity summary

- NX = 1 for time
- = 2 for source code
- = 3 for elevation
- = 4 for depth
- = 5 for height
- = 6 for Free Air anomaly
- = 7 for Bouguer anomaly
- = 8 for terrain correction
- = 9 for complete Bouguer anomaly
- = 10 for regional Free Air anomaly
- = 11 for observed gravity

JFMT = 4, MBATR Format - bathymetry

- = 1 for time
- = 2 for corrected depths in fathoms
- = 3 for corrected depths in meters
- = 4 for cumulative distance in kms
- = 5 for heading
- = 6 for speed in knots
- = 7 for uncorrected depths in fathoms
- = 8 for uncorrected depths in meters

## 4th card continued

---

JFMT = 5, CALCM Format - magnetic field

- NX = 1 for time  
= 2 for calculated regional field  
= 3 for anomalous field  
= 4 for cumulative distance in kms  
= 5 for heading  
= 6 for speed  
= 7 for observed magnetic field

---

JFMT = 6, STATN Format - stations  
Not implemented

---

JFMT = 7, SPFMT Format (Bowin format for Univ. of Toronto  
compilation of seismic refraction data)

- NX = 1 for station number  
= 2 for height  
= 3 for mantle velocity  
= 4 for depth to mantle  
= 5 for crustal thickness  
= 6 for average crustal velocity (CRVN) (Nafe-Drake)  
= 7 for column weight (WETN) using Nafe and Drake  
velocity/density relation  
= 8 for column weight (AVWTN) using average crustal density  
= 9 for CRVW Same as 6 to 8 but using Woollard's  
= 10 for WGTW velocity/density relation  
= 11 for AVWTW

---

JFMT = 8, World Seismicity Format

If SSW(16) = 1, then NX value is ignored and ANOV3 plots a  
spot whose type and size depends upon depth and  
magnitude of earthquake

If SSW(16) = 0

- NX = 1 for date (month, day, year)  
= 2 for depth in kilometers  
= 3 for magnitude

---

JFMT = 9, Active Volcanoes

NX = 1 for region code number (from IVA Catalog of Active  
Volcanoes of the World)

= 2 for height in meters

= 3 for volume and page ((IPT\*1000)+IPAGE)

\*7th card format (A4) for GSUM formatted data only  
Column 1-4 input tape serial number, one per card, as input tapes. Last tape serial number card must have  
EITP in columns 1-4, to signal end of input tape serial  
numbers (calls subroutine MOUNT).

## 4th card (Contd.)

---

JFMT = 10, Heat Flow

(For key to items 1, 6, and 7, see Simmons and Horai, Journ. Geophys. Res., Vol. 73, p. 6608-6629, 1968)

- NX = 1 for catalog sequence number
- = 2 for depth
- = 3 for heat flow
- = 4 for gradient
- = 5 for conductivity
- = 6 for classification code for station
- = 7 for reference number
- = 8 for year

---

JFMT = 11, Lunar Data

NX = No options implemented yet

---

JFMT = 12, User supplied format. Dummy

Subroutine GETX is in library. User supplies his own as a binary or source deck with job. The following conventions must be followed:

- If NX = 0 no annotations will take place.
- If NX = 1 program will annotate with time.

## 5th card Format (4I5)

## Column

5	KPT = 1 chart magnification factor (usually 1)
10	KHT = Annotation character size in integer multiples of 0.07 inch (usually 1)
15	ICTYP= 0 for non-integer degree chart boundaries = 1 for integer degree chart boundaries
19,20	IDEC = variable for decimal point in annotation of plotted points = N, for N DIGITS to right of decimal point = 0 for decimal point only =-1 for suppressing decimal point

6th card Format (4I5) Values are negative for west and south  
If ICTYP = 1 (integer degree boundaries)

## Column

1-5	ITOP = Top boundary of chart
6-10	IBOT = Bottom boundary of chart
11-15	ILEFT = Left boundary of chart
16-20	IRIGT = Right boundary of chart

If ICTYP = 0 (non-integer degree), then enter CHART boundaries on 4 cards in degrees and minutes Format (I5,F10.5)  
Be sure sign of the minutes agrees with the sign of degrees (e.g., -33-30.0) \*

DATA:

Data in specified format are loaded in device having unit reference number 1.

OUTPUT:

## Printer:

Listing of inputs

List of dates outside of chart boundaries ("OOB") if SSW(9) is up

List of dates of all data read if SSW(12) is up

## Plotter:

Mercator Charts

USAGE: See operating instructions at the end of this section

RESTRICTIONS: None

STORAGE REQUIREMENTS: 18,432<sub>10</sub> locations

\*7th card format (A4) for GSUM formatted data only.

Column 1-4 Input tape serial number, one per card, as many cards as input tapes. Last tape serial number card must have EITP in columns 1-4, to signal end of input tape serial numbers (calls subroutine MOUNT).

SUBROUTINES REQUIRED: Stored in library accounts 305, 312 and 3

GRID2, OLINE, WHR, ANOV2, RETBY, VETBY, GETC, GETF, GETG, GETM, GETS, GETST, GETH, GETP, GETV, GETY, GETL, GETX, MOUNT, STAT, ISW, TODAY, POSTAP, SPOT2, ANOV3, FIND, CALCOMP routines.

OPERATIONAL ENVIRONMENT:

Data input device - Unit reference number = 1  
12" or 30" Calcomp Plotter

OPERATIONAL CHARACTERISTICS:

Sense Switch Options: Set to zero to decline option

- SSW(0) = 1 to draw only the grid
- SSW(1) = 1 to delete drawing NDEG lines (in GRID2)
- SSW(2) = 1 if next plot will be on the same grid as this plot, sets pen back at origin
- SSW(3) = 1 to only annotate date at change of day
- SSW(4) = 0 for no mark at data point  
= 1 for plotting a circle around data point  
= 2 for plotting a dot at data point
- SSW(5) = 0 to make degree annotations inside grid (character size 0.07")  
= 1 to make degree annotations outside grid (character size 0.21")  
= 2 to make degree annotations outside grid (character size 0.35")
- SSW(6) For multiplot runs, = 1 will put on EOF between plots. Useful to PDP-5 operator for restarting in the event of mechanical malfunction of pen
- SSW(7) = 0 to annotate on right side of track  
= 1 to annotate on left side of track
- SSW(8) = 1 to suppress plotting of grid
- SSW(9) = 1 to list points out of bounds on line printer
- SSW(10) = 1 to call subroutine MOUNT which reads serial number of input tape; not used for GSUM formatted data.
- SSW(11) = 1 to annotate data points alternately on left and right side of track
- SSW(12) = 1 to list date of data just read for identification
- SSW(13) = 1 if two or more plots are to be made from the same file and this is not the last plot. Backs tape up to beginning of file and reinitializes program.

## OPERATIONAL CHARACTERISTICS (Contd.)

- SSW(16) = 1 to plot a spot for seismicity data whose type and size depends upon the depth and magnitude of the earthquake (ANOV4)
- SSW(17) = 0 (seismicity) plots an x for pre-1961 data. Depth and magnitude data pre-1961 are limited. For these points, ANOV4 normally uses a symbol which does not vary in size  
= 1 ANOV4 will try to plot varying sized symbols for all data, including pre-1961  
= 9 will not plot pre-1961 data at all  
This sense switch is used only if SSW(16)=1
- SSW(18) = 0 to make annotation at right angles to incremental track (subroutine ANOV2)  
= 1 to make annotations horizontally  
= 2 to make annotations vertically  
= 3 to invert annotations for headings 180 to 269  
= 4 to annotate either horizontally or vertically depending on direction of track
- SSW(19) = 0 for earth meridional parts from Bowditch  
= 1 for meridional parts for spherical planet
- SSW(20) = N, (seismicity) for additional size increment in plotting symbols for all data points (ANOV4). (Only if SSW(16)=1)
- SSW(21) = N, (seismicity) for size factor by which plotting symbols will vary according to magnitude. If N = 0, then ANOV4 sets N = 2. (Only if SSW(16) = 1)
- SSW(27) = 1 for GSUM data to suppress rewind input tape at start
- SSW(30) = 1 to read GSUM from 2 cards
- SSW(32) = 1 to read SPFMT from 2 cards
- SSW(40) = 1 to process GSUM with BOUNDS using DLT
- SSW(42) = 1 to read SEISMICITY data in blocked format
- SSW(60) = 1 to process GSUM data only with IFFC = 4
- SSW(61) = 1 to replace GSUM values with averaged values for FA, BG, ELEV, LAT, LONG
- SSW(71) = 1 to annotate every two hours on the hour only

Program Flow:

Tape advances to start date. Program initialization choices are made, plotter draws and annotates Mercator grid, and then data in appropriate format are read and plotted one record at a time if within chart boundaries. If more than one plot is being made the program can be restarted using SSW(13), or by using SSW(6) and by putting a RUN and DATA card and continue with a new set of data cards.

TIMING:

Two to twenty minutes depending upon size of chart, number of intermediate degree lines plotted, and amount of data plotted and annotated.

ERROR MESSAGE DIAGNOSTIC:

<u>Message</u>	<u>Cause</u>	<u>Action</u>
OOB    day, month, year, time	Date point is out of chart boundaries, and SSW(9) is up	Record is skipped, program continues
EOF    day, month, year, time	End of file found on magnetic tape	Job ends or con- tinues to next plot if any
PARITY ER    day, month, year, time	Parity error found	Record is skipped, program continues
FMT ER    day, month, year, time	Unidentified error found	Record is skipped, program continues

PROGRAMMER: Carl Bowin and Hartley Hoskins

ORIGINATOR: Carl Bowin

DATE: Version of 19 October 1972

REFERENCES: Meridional parts calculated from formula given  
on page 1186 of Bowditch, "American Practical  
Navigator", 1962 corrected reprint, Govt.  
Printing Office, Washington, D. C. 0. 0.  
Publ. No. 9.







# SEISMICITY CHARTS WITH VARYING SYMBOLS


The type of symbol is determined by depth; size varies with magnitude.  
(Subroutine ANOV4, version 15 Apr. 1975)


Depth


M<sub>b</sub> MAGNITUDE

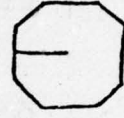
Sense switches 20 and 21 determine the size variables for each chart. In this example, both sense switches were left blank. The size increment is then automatically set equal to 2.

	- less than 70 km
	- 70 to 150 km
	- 150 to 300 km
	- 300 to 500 km
	- greater than 500 km
	- pre 1961 data

 - less than 4.5

 - 4.5 to 5.5

 - 5.5 to 6.5

 - greater than 6.5

21 December 1976

NAME: CHARTG

TYPE: Program

PURPOSE: Plot GSUM, GCON, GABS on Mercator charts

SOURCE LANGUAGE: XEROX EXTENDED FORTRAN IV

MACHINE: Sigma-7

PROGRAM CATEGORY: Graphical Display

DESCRIPTION: Plots Mercator chart at specified scale, plots and annotates with specified parameter value.

INPUT:

Input formats: GSUM, GCON, GABS

Plotter registration: Program plots a 1/2 inch fiducial square in lower right corner of chart. If sides are offset, there has been a loss of registration.

Parameter Cards

Card 1 Label - up to 80 characters written vertically on left margin of chart (FORMAT(20A4))

Card 2 Sense Switch Options - (FORMAT(80I1))  
put option in card column corresponding to sense switch (SSW( $\emptyset$ ) in column 80)

SSW( $\emptyset$ ) = 1 to only draw grid (no input data read)  
SSW(1) = 1 to delete all intermediate (NDEG) grid lines  
SSW(2) = 1 next plot will be on the same grid  
SSW(3) = 1 to annotate only at change of day  
SSW(4) = 1 to plot a circle around data point  
          = 2 to plot a dot at the data point

21 December 1976  
CHARTC  
Page 2

INI:UT (continued):

SSW(5) =  $\emptyset$  degree annotation inside grid  
           (character size = 0.07")  
           = 1 degree annotation outside grid  
           (character size = 0.21")  
           = 2 degree annotation outside grid  
           (character size = 0.35")  
 SSW(6) = 1 puts EOF between plots  
 SSW(7) =  $\emptyset$  annotate on left side of track  
           = 1 annotate on right side of track  
 SSW(8) = 1 to suppress plotting of grid  
 SSW(9) = 1 list date and time of data out of bounds  
 SSW(10) = 1 to call mount to read serial  
           number of input tape  
 SSW(11) = 1 to annotate data points alternately  
           on left and right side of track  
 SSW(12) = 1 to list date of data just read  
 SSW(18) =  $\emptyset$  annotations at right angles to track  
           = 1 annotate horizontally  
           = 2 annotate vertically  
           = 3 invert annotations for headings  
           between 180 to 269  
           = 5 to do no annotation  
 SSW(19) =  $\emptyset$  for earth meridional points from  
           Bowditch  
           = 1 for meridional points for spherical  
           planet  
 SSW(25) = 1 to call MOUNT for input tape  
           serial number  
 SSW(30) = 1 to read GSUM from punched-cards  
 SSW(40) =  $\emptyset$  process unblocked GSUM (no DLT)  
           = 1 process blocked GSUM (no DLT)  
           = 2 process blocked GSUM (with DLT)  
 SSW(46) =  $\emptyset$  process with bounds  
           = 1 to make no check on bounds  
 SSW(71) = 1 to annotate every two hours on the  
           hour

Card 3    START/END Dates    (Format(2(3I2,I4,1x),3I5))

Column

1,2	Start Day	(ISTDA)
3,4	Start Month	(ISTMO)
5,6	Start Year	(ISTYR)
7,10	Start Time	(ISTHM)
16,17	End Day	(IENDA)
18,19	End Month	(IENMO)
20,21	End Year	(IENYR)
22,25	End Time	(IENHM)

21 December 1976  
CHARTG  
Page 3

INPUT (continued):

Card 4

(Format (F10.0,3I5,1X,A4,5I5))

Column

1-10	Inches/one degree of longitude	(SINCH)
15	= 1 to draw track	(ITRK)
20	= N to plot every nth point	(LCNT)
25	= N to draw every nth grid line	(NDEG)
27-30	= number of plot	(NUMPL)
35	= N to annotate every nth plotted point	(NPTA)
39,40	= 3 to use GSUM	(JFMT)
	= 13 to use GCON	
	= 14 to use GABS	
44,45	= N to annotate with nth variable (see following table)	(NX)
50	= N to output N files on one grid	(NFILE)

Table for Selecting NX

	JFMT = 3 (GSUM)	13 (GCON)	14 (GABS)
NX			
0	omits annotation	omits annotation	omits annotation
1	time		
2	source code		
3	elevation		
4	depth	depth	average free air
5	height	height	average elevation
6	free air anomaly	free air anomaly	central free air
7	Bouguer anomaly	Bouguer anomaly	central elevation
8	terrain corrections	abstracted free air	number of points
9	complete Bouguer	abstracted height	
10	regional free air	average free air	
11	observed gravity	average height	

Card 5

(Format (4I5))

Column

5	chart magnification factor (KPT)
10	annotation character size (KHT) in integer multiples of 0.07 inch
15	= 0 for non-integer chart boundaries (ICTYP) = 1 for integer chart boundaries
19,20	= N for N digits to right of decimal point = 0 for decimal point only = -1 for suppressing decimal point

21 December 1976

CHARTG

Page 4

INPUT (continued):Card 6

(Format(4I5))

If ICTYP = 1 integer degree boundaries (Format(4I5))

## Column

1-5	Top boundary
6-10	Bottom boundary
11-15	Left boundary
16-20	Right boundary

If ICTYP = 0 non-integer boundaries (Format(I5,F10,5))

Enter one card each for top, bottom, left, right  
in degrees and minutes.

VALUES ARE NEGATIVE FOR WEST AND SOUTH

Remaining Cards

The remaining cards depend on if DLT's are used  
and if mount is called.

If neither are used, there are no more cards.

If only mount is called, then there is a card for  
each input tape of the form.

## Column

1-4	mag tape serial number (ITAPE)
-----	--------------------------------

And after all tape serial numbers there is a card with  
EITP from an ID. This signifies end of input tapes.

If D.L.T.'s are used, the D.L.T. deck is inserted  
immediately after the MOUNT serial number card for the  
appropriate tape.

OUTPUT:

Printer: Listing of input parameters

Plotter: Mercator charts

21 December 1976  
CHARTG  
Page 5

USAGE: See operating instructions

RESTRICTIONS: None

STORAGE REQUIREMENTS: 1349<sub>10</sub> locations

SUBROUTINES REQUIRED:

Stored in library accounts 456, 305, 312, and 3

GRID2, OLINE, WHR, ANOV2, RETBY, VETBY, GETGS, MOUNT, STAT,  
ISW, TODAY, POSTAP, SPOT2, ANOV3, FIND, CALCOMP routines

OPERATIONAL ENVIRONMENT:

9-track tape drive, card reader, line printer, plotter

OPERATIONAL CHARACTERISTICS:

Program Flow:

Tape advances to start date. Program initialization choices are made, plotter draws and annotates Mercator grid, and then data in appropriate format are read and plotted one record at a time if within chart boundaries.

TIMING: About 1000 pts plotted per minute if the DLT is in use.

ERROR MESSAGE DIAGNOSTIC:

<u>Message</u>	<u>Cause</u>	<u>Action</u>
OOB day, month, year, time	Data point is out of chart boundaries, and SSW(9) = 1	Record is skipped, program continues
EOF day, month, year, time	End of file found on magnetic tape	Job ends or continues to next plot if any
PARITY ER day, month, year, time	Unidentified error found	Record is skipped, program continues

21 December 1976  
CHARTG  
Page 6

PROGRAMMER: Carl Bowin, Hartley Hoskins, J.M. Monget

ORIGINATOR: Carl Bowin

DATE: May 1973

REFERENCES:

Meridional parts calculated from formula given on  
page 1186 of Bowditch, "American Practical Navigator",  
1962 corrected reprint, Govt. Printing Office,  
Washington, D.C. O.O. Publ. No. 9.

NAME: CONV67

TYPE: Main Program

PURPOSE: Convert gravity data to 1967 Geodetic Reference System and the new basic value of gravity at Potsdam, 981260, mgals.

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Data Processing

DESCRIPTION:

CONV67 converts gravity data, in GSUM format and blocked by 50, from the 1930 gravity formula and Potsdam gravity value to the 1967 Geodetic Reference System and new Potsdam gravity values. The program makes use of the Asynchronous I/O available in XDS extended Fortran IV.

INPUT:

- a) GSUM - blocked by 50 at 1930 datum (IREC=1)

OUTPUT:

- a) GSUM - blocked by 50 at IGSN71 datum (IREC=2) and referenced to International Gravity Formula 1967.

USAGE:

```
!JOB
!MESSAGE (Mag tape info)
!ASSIGN F:1, (DEVICE,9T), (SN,XXXX), (IN), (TRIES,10)
!ASSIGN F:2, (DEVICE,9T), (SN,XXXX), (OUT), (TRIES,10)
!OLAY (BI), (UNSAT, (312), (305), (456), (3))
!RUN
```

RESTRICTIONS:

- a) uses only tapes blocked by 50
- b) must have GINTF (theoretical gravity function) for 1967 datum.

CONV67 continued

SUBPROGRAMS REQUIRED: GINTF

OPERATIONAL CHARACTERISTICS:

SENSE SWITCH OPTIONS - not applicable

PROGRAM FLOW

Using BUFF IN, BUFF OUT, ENCODE and DECODE, CONV67 performs asynchronous I/O while converting observed gravity, Free-Air anomaly and Bouguer anomaly.

ERRORS AND DIAGNOSTIC MESSAGES:

Waiting for Input - the processing has halted temporarily while a block of data is read into memory

Waiting for Output- the processing has halted temporarily while a block of data is written from memory

End of File on ITAPE - end of file mark encountered on input tape

End of File on JTAPE - end of reel foil encountered on output tape, no reel change will be made.

Buffer In Error - a read error has occurred but it is not fatal and processing will continue.  
Probably will result in some lost records.

Buffer Out Error- a write error has occurred but it is not fatal and processing will continue.  
Probably will result in some lost records.

PROGRAMMER: Lee Gove

ORIGINATOR: Carl Bowin

DATE: 1 December 1973

REFERENCES:

- 1) Geodetic Reference System 1967, Bureau Central De l'association Internationale de Geodesie, 1967.

NAME: CR2G

TYPE: Main Program

PURPOSE: Converts land gravity meter counter readings  
to observed gravity values

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran

PROGRAM CATEGORY: Data Processing

DESCRIPTION:

Takes input of a counter reading and converts the counter reading to gravity, also inserts drift and tidal corrections. The program lists the data for each station, punches cards for sorting, and writes a GSUM format file with anomalies calculated in reference to IGF 1930.

INPUT:

Card 1:

Conversion tables for the counter reading to relative milligal values (I2, F7.2) 70 cards

Card 2: Sense switches (8011)

ISW(1) = 0 for printed output of computed values for each station  
= 1 for suppression of printed output  
ISW(2) = 0 to punch output for gravity description program (GDS)  
= 1 for suppression of punched output  
ISW(4) = 0 to output FILE TWO in GSUM format in preparation  
for sorting  
= 1 to suppress output into FILE TWO  
ISW(5) = 0 for meter drift correction  
= 1 suppression of drift correction

Card 3: IGM(1), IGM(2), DRFTCO, LSRC, IELC, IGC (2A4, 2X, F10.5, 3I5)

IGM - gravity meter used (e.g. L&R G-18)

If these are both blank, the type of gravity meter will be set to the default value of 'L&R G-18'

DRFTCO - The correction factor for drift of the gravity meter

If this is blank, or set to 0.0, a drift variation of 0.003 mgal/day is assumed (default value)

LSRC - Source code of GSUM output. Default value is 006 - the source for the G-18 meter.

IELC - Elevation code for GSUM output. Default value is 09.

IGC - Gravity meter code for GSUM output. Default value is 01.

CR2G (continued) page two

INPUT (continued)

These are followed by groups of individual station counter reading cards. Each group is headed by three cards:

- Card A: BASEG(1),BASEG(2) (F3.0,F6.2)  
The absolute gravity value for the reference station
- Card B: DENSE (F4.2)  
The assumed crustal density to be used in calculation of the Bouguer anomaly
- Card C: Counter reading card for the reference station. Drift is computed starting with the date on this card.
- Card D: Counter reading cards for those stations which will be referenced to the station(card C)  
Counter reading cards have the following format.  
(format of 17 May 1966):  
Station number (I4), Day (I2), Month (I2), Year (I2),  
Time (I4), Counter reading (F8.3), Latitude degrees (I2), Latitude minutes (F5.2), North or South (A1),  
Longitude degrees (I3), Longitude minutes (F5.2),  
West or East (A1), Elevation F7.1), Time Zone (I2),  
and Description (32A1).
- Card E: Either a counter reading card with all zeros (or blanks) except for the year value (card columns 9 and 10) - signals the end of a group of stations. Program then tries to read a new absolute gravity value (card A above)
- OR A card with all zeros (or blanks) - signals the end of input data.

OUTPUT:

- A. Unless sense switch (1) equals 1, records of the following format will be output to the line printer, along with a page heading.

STAT	= Station number
DATE	= Day, month, year, e.g. 10 Dec. 1970 becomes 101270
TIME	= Hour, minute
LAT	= Latitude
LONG	= Longitude
ELEV	= Elevation
CR	= Counter reading
RELV	= Relative value of gravity to counter reading
DIFF	= Difference of gravity between two readings
OBSG	= Observed gravity
GFREE	= Free-air gravity
BOUG	= Bouguer gravity
CLS	= Tidal correction
HONK	= Honkasolo correction

CR2G (continued) page Three

OUTPUT (continued)

TZONE = Time zone corresponding to time  
GDATE = Converted GMT date and time  
DAYS = Days into the year  
TDIFF = Time difference from origin  
DRIFT = Drift correction that is being applied

- B. Unless sense switch (4) equals 1, a file in GSUM format will be output to unit number 2 in preparation for sorting.
- C. Unless sense switch (2) equals 1, cards will be punched for input to gravity description program.

NOTE: A card is not punched for the reference station

RESTRICTIONS: 1) CAUTION: If the drift of the meter is positive the value of DRFTCO must be negative.

Note also that if a value of 0.00 is entered for DRFTCO, a value of 0.003 will be assumed.

2) A maximum of 9000 cards can be input

STORAGE REQUIREMENTS: 1010 decimal words

SUBPROGRAMS REQUIRED: CDATE, CHGMT, GINTF, M2DY, TIDAL

TIMING: Unknown

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: C. Bowin, J. Wolfe, S. Abbot

ORIGINATOR: C. Bowin

DATE: 1 August 1975

NAME: CR2G67

TYPE: Main Program

PURPOSE: Converts land gravity meter counter readings  
to observed gravity values

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran

PROGRAM CATEGORY: Data Processing

DESCRIPTION:

Takes input of a counter reading and converts the counter reading to gravity, also inserts drift and tidal corrections. The program lists the data for each station, punches cards for sorting, and writes a GSUM format file with anomalies calculated in reference to International Gravity Formula 1967.

INPUT:

Card 1:

Conversion tables for the counter reading to relative milligal values (I2, F7.2) 70 cards

Card 2: Sense switches (8011)

ISW(1) = 0 for printed output of computed values for each station  
= 1 for suppression of printed output  
ISW(2) = 0 to punch output for gravity description program (GDS)  
= 1 for suppression of punched output  
ISW(4) = 0 to output FILE TWO in GSUM format in preparation  
for sorting  
= 1 to suppress output into FILE TWO  
ISW(5) = 0 for meter drift correction  
= 1 suppression of drift correction

Card 3: IGM(1), IGM(2), DRFTCO, LSRC, IELC, IGC (2A4, 2X, F10.5, 3I5)

IGM - gravity meter used (e.g. L&R G-18)

If these are both blank, the type of gravity meter will be set to the default value of 'L&R G-18'

DRFTCO - The correction factor for drift of the gravity meter

If this is blank, or set to 0.0, a drift variation of 0.003 mgal/day is assumed (default value)

LSRC - Source code of GSUM output. Default value is 006 - the source for the G-18 meter.

IELC - Elevation code for GSUM output. Default value is 09.

IGC - Gravity meter code for GSUM output. Default value is 01.

CR2G67 (continued) page two

# INPUT (continued)

These are followed by groups of individual station counter reading cards. Each group is headed by three cards:

- Card A: BASEG(1), BASEG(2) (F3.0, F6.2)  
The absolute gravity value for the reference station
- Card B: DENSE (F4.2)  
The assumed crustal density to be used in calculation of the Bouguer anomaly
- Card C: Counter reading card for the reference station. Drift is computed starting with the date on this card.
- Card D: Counter reading cards for those stations which will be referenced to the station (card C)  
Counter reading cards have the following format.  
(format of 17 May 1966):  
Station number (I4), Day (I2), Month (I2), Year (I2), Time (I4), Counter reading (F8.3), Latitude degrees (I2), Latitude minutes (F5.2), North or South (A1), Longitude degrees (I3), Longitude minutes (F5.2), West or East (A1), Elevation F7.1), Time Zone (I2), and Description (32A1).
- Card E: Either a counter reading card with all zeros (or blanks) except for the year value (card columns 9 and 10) - signals the end of a group of stations. Program then tries to read a new absolute gravity value (card A above)
- OR A card with all zeros (or blanks) - signals the end of input data.

# OUTPUT:

- A. Unless sense switch (1) equals 1, records of the following format will be output to the line printer, along with a page heading.

STAT = Station number  
DATE = Day, month, year, e.g. 10 Dec. 1970 becomes 101270  
TIME = Hour, minute  
LAT = Latitude  
LONG = Longitude  
ELEV = Elevation  
CR = Counter reading  
RELV = Relative value of gravity to counter reading  
DIFF = Difference of gravity between two readings  
OBSG = Observed gravity  
GFRF = Free-air gravity  
BOUG = Bouguer gravity  
CLS = Tidal correction  
HONK = Honkasolo correction

CR2G67 (continued) page three

OUTPUT (continued)

TZONE = Time zone corresponding to time  
GDATE = Converted GMT date and time  
DAYS = Days into the year  
TDIFF = Time difference from origin  
DRIFT = Drift correction that is being applied

- B. Unless sense switch (4) equals 1, a file in GSUM format will be output to unit number 2 in preparation for sorting.
- C. Unless sense switch (2) equals 1, cards will be punched for input to gravity description program.

NOTE: A card is not punched for the reference station

RESTRICTIONS: 1) CAUTION: If the drift of the meter is positive the value of DRFTCO must be negative.

Note also that if a value of 0.00 is entered for DRFTCO, a value of 0.003 will be assumed.

2) A maximum of 9000 cards can be input

STORAGE REQUIREMENTS: 1010 decimal words

SUBPROGRAMS REQUIRED: CDATE, CHGMT, GINTF, M2DY, TIDAL

TIMING: Unknown

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: C. Bowin, J. Wolfe, S. Abbot

ORIGINATOR: C. Bowin

DATE: 1 August 1975

NAME: CRWT3

TYPE: Main Program

PURPOSE: To calculate the pressure at the base of a crustal column ( $\text{Kg/cm}^2$ ).

MACHINE: Sigma 7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Mathematical (equation solving)

DESCRIPTION:

Reads U. of Toronto World Seismic Refraction Compilation at W.H.O.I. SPFMT format. Data may input on cards or mag tape.

Table look-up values for the conversion of compressional seismic velocity to density are entered during initialization. Compensation depth (e.g. 40 km) is also entered during initialization. Seismic refraction data of SPFMT format is read and the pressure at the compensation depth is then calculated. Crustal thickness, average crustal velocity and depth to mantle are also calculated and output in the SPFMT format.

INPUT:

Card 1: Sense switch options: Put sense switch 0 in column 80.

ISW(0) = 1 to list intermediate values for testing  
 ISW(26)=1 to output on line printer only  
 ISW(32) =1 to read SPFMT data from two cards per record  
 ISW(33) =1 to write SPFMT data on two cards per record

Card 2: ICTAB, DCOMP (I5,F10.0)  
 ICTAB = 0 for Nafe-Drake Density Table  
       = 1 for Woollard Density table  
 DCOMP = depth of compensation (Km).

Card 3: Density table cards (10F8.3) 10 values per card

Card(s) 4: (optional) SPFMT data cards, if data is on cards

Card 5: !EOD if data is on cards

CRWT3

OUTPUT: Data can be output either to mag tape or cards, depending on sense switches and control cards. If ISW(0) = 1, values read in and calculated are listed with annotation on the line printer.

USAGE:

Assign F:1 to input device; F:2 to output device

RESTRICTIONS:

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: EVIL, EXIT, ISW, NAVIN, PINOT, STAT, TODAY

TIMING: Undetermined

ERRORS AND DIAGNOSTICS: If ISW(0) = 1, the program outputs annotated lists of values read and calculated.

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 2 October 1974

NAFE-DRAKE EXPERIMENTAL RELATIONSHIP  
(In Talwani, Sutton, and Worzel, 1959  
JGR, v. 14, No. 10, p. 1548)  
(Picks by C. Bowin)

$v_p$	$\rho$	$v_p$	$\rho$	$v_p$	$\rho$
1.0	0	4.0	2.39	7.0	3.04
1.1	0	4.1	2.41	7.1	3.07
1.2	0	4.2	2.425	7.2	3.10
1.3	0	4.3	2.44	7.3	3.13
1.4	0	4.4	2.45	7.4	3.16
1.5	1.47	4.5	2.48	7.5	3.19
1.6	1.66	4.6	2.50	7.6	3.22
1.7	1.73	4.7	2.52	7.7	3.25
1.8	1.80	4.8	2.53	7.8	3.28
1.9	1.86	4.9	2.55	7.9	3.31
2.0	1.92	5.0	2.57	8.0	3.34
2.1	1.98	5.1	2.59	8.1	3.38
2.2	2.01	5.2	2.61	8.2	3.42
2.3	2.03	5.3	2.62	8.3	3.46
2.4	2.06	5.4	2.64	8.4	3.49
2.5	2.09	5.5	2.66	8.5	3.525
2.6	2.11	5.6	2.68	8.6	3.56
2.7	2.13	5.7	2.70	8.7	3.59
2.8	2.15	5.8	2.72	8.8	3.63
2.9	2.18	5.9	2.74	8.9	3.67
3.0	2.21	6.0	2.77	9.0	3.71
3.1	2.23	6.1	2.80	9.1	3.74
3.2	2.24	6.2	2.83	9.2	3.78
3.3	2.26	6.3	2.85	9.3	3.82
3.4	2.28	6.4	2.87	9.4	3.85
3.5	2.30	6.5	2.90	9.5	3.88
3.6	2.32	6.6	2.93	9.6	3.91
3.7	2.34	6.7	2.95	9.7	3.95
3.8	2.36	6.8	2.975	9.8	3.99
3.9	2.375	6.9	3.01	9.9	4.02
				10.0	4.06

VEL-DENS RELATIONSHIP  
(From Woollard (1959))

<u>VEL.</u> Km/sec.	<u><math>\rho</math></u> gm/cm <sup>3</sup>	<u>VEL.</u> Km/sec	<u><math>\rho</math></u> gm/cc	<u>VEL.</u> Km/sec	<u><math>\rho</math></u> gm/cm <sup>3</sup>
1.0	1.62	4.0	2.61	7.0	3.06
1.1	1.62	4.1	2.62	7.1	3.09
1.2	1.63	4.2	2.62	7.2	3.12
1.3	1.66	4.3	2.62	7.3	3.15
1.4	1.69	4.4	2.62	7.4	3.17
1.5	1.74	4.5	2.62	7.5	3.20
1.6	1.81	4.6	2.62	7.6	3.23
1.7	1.88	4.7	2.63	7.7	3.25
1.8	2.06	4.8	2.64	7.8	3.28
1.9	2.18	4.9	2.65	7.9	3.31
2.0	2.27	5.0	2.66	8.0	3.33
2.1	2.34	5.1	2.67	8.1	3.36
2.2	2.39	5.2	2.68	8.2	3.39
2.3	2.42	5.3	2.69	8.3	3.42
2.4	2.45	5.4	2.70	8.4	3.45
2.5	2.49	5.5	2.71	8.5	3.47
2.6	2.51	5.6	2.73	8.6	3.50
2.7	2.53	5.7	2.74	8.7	3.53
2.8	2.55	5.8	2.76	8.8	3.55
2.9	2.56	5.9	2.79	8.9	3.58
3.0	2.56	6.0	2.81	9.0	3.61
3.1	2.57	6.1	2.83		
3.2	2.58	6.2	2.85		
3.3	2.59	6.3	2.88		
3.4	2.59	6.4	2.90		
3.5	2.60	6.5	2.93		
3.6	2.60	6.6	2.96		
3.7	2.60	6.7	2.99		
3.8	2.61	6.8	3.01		
3.9	2.61	6.9	3.04		

NAME: DMABLK

TYPE: Main Program

PURPOSE: Converts blocked DMA format data to blocked GSUM  
format data

MACHINE: Sigma-7

SOURCE LANGUAGE Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:

DMABLK is a modification of DMA which, in turn, is a modification of program ACTG3.

DMABLK reads DMA data blocked by 50 and converts data with elevation codes 1 and 3 to GSUM formatted data blocked by 50. Source code and beginning sequence number are entered at run time; sequence number is output in station number field. Data records with elevation codes other than one and three are output to another tape in DMA format for further processing.

INPUT:

Card 1: NSEQ (I10) - starting sequence number

Card 2: ISORC (I5) - source code for this data

ERRORS AND DIAGNOSTICS:

'WAITING FOR I/P' - input buffer not yet filled when checked

'END OF FILE ON ITAPE' - end of file found on input tape

'NUFFER IN ERROR' - input buffer error detected by ICHECK

'WAITING FOR OUTPUT' - output buffer not yet filled when checked

'END OF FILE JTAPE' - end of reel encountered on output tape

'BAD JKEY' - end of reel encountered on output tape

'ALL DONE'

OUTPUT:

On unit reference number 2: GSUM records for elevation codes 1 and 2  
On unit reference number 3: DMA records for other elevation codes

The number of records input, records output to each output tape, and ending sequence number are output to line printer.

USAGE:

Assign F:1 to input device; F:2 to output device for GSUM records;  
F:3 to output device for 'oddball' records (elevation codes other than 1 and 3).

RESTRICTIONS: None

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: ALTD, XEROX Fortran IV Library

TIMING: about 1,000 records per minute

PROGRAMMER: Lee Gove, C. Bowin

ORIGINATOR: C. Bowin

DATE: 31 July 1975

NAME: DMAP

TYPE: Main Program

PURPOSE: Converts digitized position in inches to latitude and longitude (radians)

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:

Digitized x and y coordinate values from a digitizing table for the four corners of a map region and read by the program along with latitude and longitude of each point. From this information the angle of tilt of the map and its scale are determined. Other x and y coordinate data points are then read in and the latitude and longitude of each is calculated to a precision governed by the input value for the variable EPSIL.

INPUT:Card 1: sense switches

SSW(2) = 1 to list intermediate values  
 SSW(3) = 1 to list date and SMIN for each data point  
 SSW(5) = 1 to output SMIN only if greater than EPSIL

Card 2: ITAPE, JTAPE, EFAC, EPSIL (2I5,2F5.2)

ITAPE - unit number for input device  
 JTAPE - unit number for output device  
 EFAC - factor (0.1 to 1.00) used on iteration for estimated latitude to converge on true latitude  
 EPSIL - tolerance (in meridional parts) by which estimated latitude must match meridional parts for true latitude.

Card 3, 4, 5, and 6: ICODE, XC(J), YC(J), N1, N2, LAT(J) LONG(J)  
 (I1,1X,F5.3,1X,F5.3,3I3,I5)

ICODE = 9 for cards 3, 4, 5 and 6 for initialization  
 J in do loop is = 1 for bottom left corner,  
 then 2,3,4 counter clockwise around map corners  
 XC(J) = X coordinate value in inches  
 YC(J) = Y coordinate value in inches  
 N1 = not used  
 N2 = not used  
 LAT(J) = Latitude  
 LONG(J) = Longitude

Card 7: ICODE, XP, YP, NDA, NMO, NYR, NHM  
(I1, 1X, F5.3, 1X, F5.3, 3I3, I5)

ICODE = 5 for data points  
 XP = X coordinate value in inches  
 YP = Y coordinate value in inches  
 NDA = Day  
 NMO = Month  
 NYR = Year  
 = 0 on terminator card to indicate last  
 data point has been processed.  
 NHM = Hours and minutes (24 hours clock)

**OUTPUT:**

Outputs record containing latitude and longitude  
for each input data point.

USAGE: Assign input and output devices compatible with ITAPE  
and JTAPE values entered on card 2.

RESTRICTIONS: None

STORAGE REQUIREMENTS: Undetermined

SUBPROGRAMS REQUIRED: CALSC, DMTOR, ISW, PARTM, RTODM

TIMING:

ERRORS AND DIAGNOSTICS: Undetermined

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 26 September 1975

NAME: DMOD

TYPE: Main Program

PURPOSE: To punch digitized polygon points for crustal models digitized on a digitizing table

MACHINE: Sigma 7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format Conversion

DESCRIPTION:

This program converts the coordinates for polygon points punched by a digitizing table to the correct format for use in a TALPLOT run and punches the cards necessary for all polygon points using origin and scale factors input to this program at run time. Input and output are always on cards. For instructions in the use of the digitizing table, see comments under USAGE.

The program initializes by reading sense switches, scale factors and coordinates origin. It sets the origin to the coordinates of first digitized point entered, uses the second digitized point to establish a horizontal reference line and then calculates X and Y distances of all points from the origin using the input scale factors. It punches cards with the adjusted and scaled X and Y coordinates along with the identification number of each polygon; one card for each polygon in which the point occurs.

#### INPUT:

Card 1: Sense Switch Options: (8011)  
           ISW(1) = 0 for second point to right (+) of origin  
                   = 1 for second point to left (-) of origin

Card 2: XFAC,YFAC,XORG,YORG (4 F10.0)  
           XFAC = scale factor in X direction (km/in)  
           YFAC = scale factor in Y-direction (km/in)  
           XORG = X-coordinate of origin of model (km)  
           YORG = Y-coordinator of origin of model (km)

The following cards are all punched at the digitizing table

Card 3: XA,YA,IA,KP1,KP2,KP3 (2F10.3,I5,3I4)  
           XA X and Y Coordinates from digitizing table from  
           YA its origin

IA } =0 (same format as card(s) 5 below, but not used  
 KP1 } for this point)

Card(s) 5: Values from the digitizing table for the polygon points XP,YP,ICODE,KP1,KP2,KP3 (2F10.3,I5,3I4)

XP X and Y coordinates of the polygon point

YP

ICODE = 9 for last point of a polygon

= 8 for X=-3000 km

= 7 for X=+3000 km

(ICODE=7 or 8 is used in this program only. Points with

ICODE = 7 or 8 are punched by this program with ICODE = 0).

KP1 numbers of the polygons for which this point forms

KP2 a boundary. One output card will be punched for each

KP3 polygon listed here.

To indicate end of input cards, an additional polygon point card with ICODE set equal to 99 must follow the last digitized point.

OUTPUT: On line printer: the digitized points

On cards: Cards in the correct format for use in TALPLOT run. Values punched are X coordinate in km., Y coordinate in km., ICODE, and the number of the polygon for which the card was punched. Cards will usually not be in the correct order and there may be some extra cards (if the first polygon point is not the first polygon point for another polygon that it defines).

USAGE:

A crustal model is prepared which is composed of various polygons of various densities. The polygons are numbered arbitrarily, with the exception of polygon number 1, which is a water layer, and the final polygon, which must be number 99. One point of each polygon is designated the "starting point". Points define the polygons by proceeding clockwise from the starting point and ending exactly at the same point. Polygon points must be arranged in order for input to the TALPLOT program, but need not be digitized in order nor input in order to the program IMOD. Output from program IMOD must be rearranged for output to TALPLOT.

At the digitizing table, the first point digitized must be the origin. The second point is a point on the same X axis as the origin, and is used to establish the horizontal for the model. The remaining points may be digitized in any order. Before lining up a point, ICODE is set in the leftmost thumbwheel switch position on the manual entry switches. ICODE = 9 to indicate the last point of any given polygon. ICODE = 7 will punch a card at the same Y coordinate as the point under the digitizing screen, but

DMOD continued, page 3

with +3000 km as the X coordinate. ICODE = 8 creates a card with -3000 km as the X coordinate. These are used at the sides of the model to extend the edges of the polygons beyond the area for which gravity will be calculated in order to avoid an edge effect. In addition, the numbers of the polygons for which the given point delineates a boundary are set in the three pairs of thumbwheel switches to the right of the leftmost thumbwheel switch. In the DMOD program, a polygon point coordinate card is punched for each polygon number inserted here.

#### RESTRICTIONS:

- 1) When punching the first and last cards for each polygon on the digitizer, make sure that the cards read exactly the same values - otherwise the polygon will not close.
- 2) Right and down are positive on the model graph. That means that Y coordinates of polygon points below the sea surface are positive numbers.
- 3) Input and output must be on cards.

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: CALSC, EXIT, ISW

TIMING: Undetermined

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 24 July 1975

NAME: GFLDI

TYPE: Main Program

PURPOSE: Calculation of regional free-air anomaly values for a given region from spherical harmonic coefficients.

MACHINE: SIGMA 7

SOURCE LANGUAGE: FORTRAN IV

PROGRAM CATEGORY: Data Processing

DESCRIPTION:

Area bounds and increment size for region of interest are entered. GFLDI next reads spherical harmonic coefficients defining a gravitational field. The program then steps across the region defined by the input area bounds calculating the regional gravity field at each position increment. Program outputs the regional free-air anomaly value in the free-air position of SEAG1 format.

INPUT:

Card 1: Sense switches (8011)  
ISW(4) = 1 to list data on high speed printer

Card 2: ITAPE, JTAPE (215)  
ITAPE = input device number (used for input of spherical harmonic coefficients)  
JTAPE = output device number

Card 3: ITOP, IBOT, ILEFT, IRIGHT, INC (515)  
ITOP = integer degree for top area boundary  
IBOT = integer degree for bottom area boundary  
ILEFT = integer degree for left area boundary  
IRIGHT = integer degree for right area boundary  
INC = integer degree increment for do loop in defining positions at which regional free-air anomalies will be calculated.

Card 4: Spherical harmonic coefficients  
Format (I2,2X,I2,2X,E11.4,2X,E11.4) followed by 2  
!EOD card - this input can be on magnetic tape or disc by appropriate value of ITAPE, on card 2 above

OUTPUT:

On unit reference JTAPE. The data in SEAG1 format with regional free-air values in free-air field.

GFLD1

USAGE: Assign input and output devices to ITAPE and JTAPE  
values input on card 2.

RESTRICTIONS: None

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: ISW, FLD2, standard Fortran IV Library

TIMING:

ERRORS AND DIAGNOSTICS:

PROGRAMMER:

ORIGINATOR:

DATE:

NAME: GFLD2

TYPE: Main Program

PURPOSE: Calculates regional free-air gravity anomalies from spherical harmonic coefficients entered at run time for location of input GSUM records.

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Data processing

DESCRIPTION:

GFLD2 reads spherical harmonic coefficients defining a gravitational field. A regional gravity value is computed from the input coefficients at latitude and longitude locations read from input GSUM records.

Program outputs regional free-air value in regional gravity position of GSUM format.

INPUT:

Card 1: Sense switches (80I2)

ISW(5) = 1 to list intermediate values for BV, COSD on line printer (SUB.FLD2)

ISW(12)= 1 to list date identification on line printer (SUB. GINOT)

ISW(26)= 1 to output on line printer only (SUB. GINOT)

ISW(29)= 1 to process only selected source codes  
= 2 to skip selected source codes (SUB.GINOT)

ISW(30)= 1 for input data on cards (SUB.GINOT)

ISW(31)=1 for output data on cards (SUB. GINOT)

ISW(40)= 0 to process without bounds  
= 1 to process with bounds using the Data Location Table (SUB.GINOT)

ISW(60)= 1 to process only Abstracter output (SUB. GINOT)

ISW(61)= 1 to replace FA, BG, ELEV, LAT, AND LONG with averaged values (SUB. GINOT)

Card(s) 2: Spherical harmonic coefficients format (I2,2X,I2, 2X,E11.4,2X,E11.4) followed by a !EOD card

Card 3: (optional) ISRC (16I5) See example for SAO Standard Earth 1969  
If ISW(29) does not equal zero, enter here up to 16 source codes to be selected (ISW(29)=2) or skipped (ISW(29)=1).

If input is on magnetic tape:

## GFLD2

Card(s) 4: Serial number(s) of input tapes, one per card,  
in columns 1 to 4 (used by subroutine MOUNT)

Card 5: EITP in columns 1 to 4 - signals end of input tape  
serial numbers

If input is on cards:

Card(s) 4: data cards in GSUM format-two cards per record

Card 5: !EOD card

If output is to magnetic tape:

Card(s) 6: Serial number(s) of output tapes, one per  
card in columns 1 to 4 (used by subroutine MOUNT).

Card 7: EOTP in col. 1 to 4 - signals end of output tape serial  
numbers.

There will be no cards 6 or 7 if output is on cards.

OUTPUT: On unit reference no. 2 - the data in GSUM format,  
with regional free-air values in regional free-air field.

USAGE: Assign F:1 to input device; F:2 to output device

RESTRICTIONS: None

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: FLD2,GINOT,ISW, MOUNT, STAT, TODAY  
Standard Fortran IV Library

TIMING: Undetermined

ERRORS AND DIAGNOSTICS: 'EOF FOUND ON INPUT TAPE'

PROGRAMMER: A. Folinsbee, Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 5 August 1975

NAME: GRAFG2  
TYPE: Main Program  
PURPOSE: To plot one variable versus another  
MACHINE: Sigma - 7  
SOURCE LANGUAGE: FORTRAN IV  
PROGRAM CATEGORY: Graphical Display

DESCRIPTION:

GRAFG2 creates a graph, plotting one variable against another. It uses input either in GSUM (gravity summary) format, or WHOLG (WHOI lunar gravity) format. It uses latitude and longitude for processing bounds. In addition, it can use either start and end dates or altitude bounds (in lunar format) as further limits to the data processed, if desired.

INPUT:

CARD 1: LABEL (20A4)  
 information for plot label

CARD 2: sense switch settings:  
 options

ISW(0) = 1 to output values for testing  
 ISW(3) = 1 to plot Lunar gravity (calls GETL)  
 ISW(4) = 1 to read SVEC altitude bounds for  
 lunar data and process data only  
 within these altitude bounds.  
 ISW(7) = 1 to input new area bounds for next plot  
 ISW(8) = 1 to suppress plotting grid  
 ISW(10) = 1 to start a new graph  
 ISW(12) = 1 to list date identification  
 ISW(13) = 1 to anotate plot point with DATAW  
 ISW(30) = 1 to read GSUM data from cards  
 ISW(34) = 1 to read Lunar data from cards

GRAFG2 continued

-2-

CARD 3: ITAPE, NX, NY, NZ, NW, IDEC, KPT, KHT (8I5)

ITAPE = input tape device number (must agree with control cards) Should = 105 to read from cards.

NX = PLT(NX) for X variable  
 NY = PLT(NY) for Y variable  
 NZ = PLT(NZ) for Z variable  
 NW = PLT(NW) for W variable

} see page 3

IDEC = code for decimal point in annotation of DATAW

KPT = plot size factor-varies size of entire plot (should = 1 in standard plot)

KHT = character height factor (varies by multiples of 0.07)

CARD 4: XFAC, YFAC, ZFAC, WFAC, ANGB, XINC, YINC (7F10.0)

XFAC = engineering units per inch on X axis

YFAC = engineering units per inch on Y axis

ZFAC = engineering units per inch on Z axis

WFAC = engineering units per inch on W axis

ANGB = angle for DATAW anotation

XINC = spacing in decimal inches for anotation in x-direction

YINC = spacing in decimal inches for anotation in y-direction

CARD 5: TOP, BOT, DLEFT, RIGHT (4F10.0)

TOP  
 BOT  
 DLEFT  
 RIGHT

} bounds for graph in engineering units

CARD(s) 6-9: area bounds in degrees and decimal minutes, one per card (I5,F10.0)

(6) IDEG, AMIN (TOP)  
 (7) " " (BOTTOM)  
 (8) " " (LEFT)  
 (9) " " (RIGHT)

Note: The sign of the minutes must agree with the sign of the degrees (e.g. -36 -30.0).

GRAFG2 continued

-3-

Values for NX, NY, NZ, and NW depend on input format.

For GSUM format

- = 1 for KGHM (time)
- = 2 for ISORC (source code)
- = 3 for ELEV (elevation)
- = 4 for DEPTH (depth)
- = 5 for HEIGHT (both depth and elevation in the same parameter)
- = 6 for FA (free-air anomaly)
- = 7 for BG (Bouguer anomaly)
- = 8 for TC (terrain correction)
- = 9 for BGCOM (complete Bouguer anomaly)
- = 10 for RFA (regional free air)
- = 11 for GOBS (observed gravity)
- = 12 for WEIGHT/BG

For lunar data (WOLG format)

- = 1 for SVEC (vehicle distance from center of mass in km)
- = 2 for SVEC-1738.0 (vehicle distance from center of mass in km minus radius)
- = 3 for ALTL (laser altitude)
- = 4 for ((SVEC-ALTL)-1738.0)
- = 5 for AZ (azimuth)
- = 6 for SINC (inclination)
- = 7 for STAC (tangential acceleration)
- = 8 for SNAC (normal acceleration)
- = 9 for FA (free-air anomaly-radial acceleration)
- = 10 for THEOR (theoretical gravity)
- = 11 for GOBS (observed gravity)
- = 12 for ELEV (elevation of topography with reference to radius)
- = 13 for ELFL (laser altitude, with reference to radius)
- = 14 for BG (Bouguer anomaly)
- = 15 for TACEL (total acceleration)

GRAFG2 continued

-4-

CARD 10: (optional) BSVEC,TSVEC (2F10.3)  
 spacecraft altitude bounds for lunar data  
 if ISW (4) =1.  
     BSVEC - lower altitude limit  
     TSVEC - upper altitude limit

CARD 11: ISTDA, ISTMO, ISTYR, ISTHM, IENDA, IENMO,  
 IENYR, IENHM, ISKP  
 (3I2, I4, 5X, 3I2, I4, 5X, I5)

Start date for processing

ISTDA - day  
 ISTMO - month  
 ISTYR - year  
 ISTHM - time

End date for processing

IENDA - day  
 IENMO - month  
 IENYR - year  
 IENHM - time

ISKP - number of records to be skipped  
 at start of job. Much faster than  
 start date alone.

To avoid checking for start date, use a blank  
 card. Plotting will then begin with the first  
 record.

CARD 12: ITAPID (I4)  
 Serial number of input tape(s), one per card.

CARD 13: EITP in columns 1-4  
 (signals end of input tape serial numbers)  
 There will be no cards 12 and 13 if input is on cards.

CARD(s) 14:  
 additional start and end dates for processing  
 may be inserted here - last card must have  
 start date = 99 to end processing.

#### OUTPUT:

Input parameters are listed on line printer. Graph can be  
 output either to versatec or to calcomp plotter.

GRAFG2 continued

-5-

USAGE:

Any number of additional graphs may be run in the same job, by use of sense switches, and inserting additional data cards 6-11, to process data with new area bounds, or just a new start date (card 11).

RESTRICTIONS:

STORAGE REQUIREMENTS: 30 peak core pages (Core 15), on the limit card

SUBPROGRAMS REQUIRED: DMTOR, EXIT, FIND, GETG, GETL, GRIDG, ISW, NUMBER, PLOT, PLOTS, SETSKP, SKPREC, SPOT, STAT, SYMBOL, TODAY, WHERE

TIMING: Undetermined

ERRORS AND DIAGNOSTICS:

'END DATE PASSED' date

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 2 June 1975

NAME: GRAV1

TYPE: PROGRAM

PURPOSE: Converts data input at format of 8 July 1969 to GSUM format.

MACHINE: SIGMA-7

SOURCE LANGUAGE: FORTRAN IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION: GRAV1 converts data input at format of 8 July 1969 to 128 character GSUM format. Latitude and longitude are output in decimal degrees. Observed gravity can be calculated from FA anomaly, and is referenced to the IGSN-71 datum. The Bouguer anomaly is calculated, using reference density entered at run time. Anomalies may be input either in 1930 or 1967 International Gravity Formula, but all output is in 1967 IGF. If necessary, station numbers are assigned, numerically starting with first input record as 1. Resulting records can be listed on line printer, or intermediate values can be output, if desired. Data may be input and output either on cards or on magnetic tapes, depending upon control cards.

INPUT:

Card 1: Sense switch settings - Punch SSW(0) in column 80.

SSW(0) = 1 for input elevation in feet  
           = 0 for input elevation in meters

SSW(2) = 1 for input depth in fathoms  
           = 0 for input depth in meters

SSW(3) = 1 to calculate observed gravity from FA anomaly

SSW(4) = 1 for input data at Potsdam Reference System and 1930 International Gravity Formula  
           = 0 for input data at IGSN-71 datum and 1967 IGF

-2-

SSW(6) = 1 for incorporating terrain correction  
          = 0 not to use terrain correction  
SSW(8) = 1 to print values of THEO, FELEV, & TH67  
SSW(13) = 1 to assign station numbers, numerically,  
          starting with first input record as 1  
SSW(26) = 1 to output on line printer only (GINOT)  
SSW(31) = 1 to output data on two cards per record  
          (GINOT)

Card 2: CRDEN (FIO.0)  
          CRDEN - assumed crustal density

Card 3: If output is to be on mag tape, output tape serial  
          number (I4), in columns 1-4, one per card, as  
          many cards as necessary

Card 4: EOTP in columns 1-4. Signals end of output tape serial  
          numbers. Not necessary if output is on cards or  
          line printer

Card(s) 5: Data cards if input is on cards

Card 6: IEOD

OUTPUT: Assumed crustal density, as input, is output to line  
          printer. Records are output to line printer or output  
          device depending on sense switch options. Records  
          output to line printer begin with the second character  
          of the record, and do not include IREC2. In addition,  
          intermediate values for theoretical gravity may be  
          output.

USAGE: Assign F:1 to input device; F:2 to output device.

RESTRICTIONS: None

STORAGE REQUIREMENTS: 21 peak core (pages)

SUBPROGRAMS REQUIRED: AREAK, STAT, GINOT, GINTF, GI67F, ISW,  
NAVIN, EVIL, RTDM2

TIMING: Unavailable

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 27 May 1975

NAME: GSTOG

TYPE: Main Program

PURPOSE: Converts data input in SEAG1 or SEAG2 format to 128 character GSUM format.

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:

GSTOG is our standard conversion program. It inputs data in SEAG format and outputs 128-character GSUM format. It tests for invalid observed gravity and invalid free-air anomaly values. If either data parameter is invalid, that record is rejected. Checks for no depth or elevation information and if lacking sets Bouguer anomaly value to invalid code (999.0).

If IREC = 1, input data is at Potsdam system and used IGF 1930 (SEAG1 format).

If IREC = 2, input data is at IGSN71 and used IGF 1967 (SEAG2 format) and proper output will be provided at GSUM format.

INPUT:

Card 1: Sense Switch options -

SSW(26) = 1 to output on line printer only (GINOT)  
SSW(31) = 1 to output data on two cards per record (GINOT)

Card 2: ISORC, IDCOD, IELC, IGC, BIAS (4I5,F10.0)

ISORC = source code number  
IDCOD = 0 for ID by date  
          = 1 for ID by station number  
IELC = elevation code  
IGC = gravity meter code  
BIAS = gravity meter bias (in mgals)

Card 3: NFILE (I5) NFILE = number of files to be input

Card 4: EITP in columns 1-4

## GSTOG

Card 5: If output is to be on mag tape, output tape serial number (I4), in columns 1-4, one per card, as many cards as necessary.

Card 6: EOTP in columns 1-4. Signals end of output tape serial numbers. Not necessary if output is on cards or line printer.

OUTPUT:

Input parameters are output to line printer. Number of records output and number of records rejected are output to line printer.

Data records can be output on cards if desired by appropriate use of control cards and sense switches. Records may be output to line printer and if so, begin with the second character of the record and do not include IREC.

USAGE:

Assign F:1 to input device; F:2 to output device

RESTRICTIONS:

STORAGE REQUIREMENTS: 23 peak core pages (core,12)

SUBPROGRAMS REQUIRED: BICOR, EVIL, EXIT, GINOT, ISW, MCVOL, STAT, TODAY, UNPKBY

TIMING: CPU time = 12.9 min. to process 10,500 input records;  
8,400 output records.

ERRORS AND DIAGNOSTICS:PROGRAMMER:

ORIGINATOR: Carl Bowin

DATE: 10 July 1975

NAME: GSTOG67

TYPE: Main Program

PURPOSE: Converts data input in SEAG1 or SEAG2 format to 128 character GSUM format and converts Potsdam system data to IGSN71

MACHINE Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:

GSTOG is our standard conversion program. It inputs data in SEAG format and outputs 128-character GSUM format. It tests for invalid observed gravity and invalid free-air anomaly values. If either data parameter is invalid, that record is rejected. Checks for no depth or elevation information and if lacking sets Bouguer anomaly value to invalid code (999.0).

If IREC = 1, input data is at Potsdam system and used IGF 1930 (SEAG1 format). OUTPUT is at IGSN 1971.

If IREC = 2, input data is at IGSN71 and used IGF 1967 (SEAG2 format) and proper output will be provided at GSUM format.

INPUT:

Card 1: Sense Switch options -

SSW(26) = 1 to output on line printer only (GINOT)  
SSW(31) = 1 to output data on two cards per record (GINOT)

Card 2: ISORC, IDCOD, IELC, IGC, BIAS (4I5,F10.0)

ISORC = source code number  
IDCOD = 0 for ID by date  
          = 1 for ID by station number  
IELC = elevation code  
IGC = gravity meter code  
BIAS = gravity meter bias (in mgals)

Card 3: NFILE (I5) NFILE = number of files to be input

Card 4: EITP in columns 1-4

## GSTOG 67

Card 5: If output is to be on mag tape, output tape serial number (I4), in columns 1-4, one per card, as many cards as necessary.

Card 6: EOTP in columns 1-4. Signals end of output tape serial numbers. Not necessary if output is on cards or line printer.

OUTPUT:

Input parameters are output to line printer. Number of records output and number of records rejected are output to line printer.

Data records can be output on cards if desired by appropriate use of control cards and sense switches. Records may be output to line printer and if so, begin with the second character of the record and do not include IREC.

USAGE:

Assign F:1 to input device; F:2 to output device

RESTRICTIONS:

STORAGE REQUIREMENTS: 23 peak core pages (core,12)

SUBPROGRAMS REQUIRED: BICOR, EVIL, EXIT, GINOT, ISW, MCVOL, STAT, TODAY, UNPKBY

TIMING: CPU time = 12.9 min. to process 10,500 input records;  
8,400 output records.

ERRORS AND DIAGNOSTICS:PROGRAMMER:

ORIGINATOR: Carl Bowin

DATE: 10 July 1975

NAME: G3DCP

TYPE: Main Program

PURPOSE: Computes gravity anomaly (for both flat and curved planetary surfaces), potential field (for flat surfaces), and mass per unit area for a set of polygonal laminae comprising a three-dimensional crustal structure model.

MACHINE: XDS Sigma 7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Modeling

DESCRIPTION:

The program is based on a method developed by Talwani and Ewing (1960). The three-dimensional body is represented by depth contours. The depth contours are replaced by flat polygonal laminae. The gravity anomaly is evaluated for the laminae by a formula expressed in a closed form. A correction is then calculated for the curved surface and this correction is applied to the flat anomaly to arrive at a curved anomaly (see Bowin et al., in press). The anomaly for the entire body is then obtained by a numerical integration for the values of the individual laminae. The Z-axis is chosen positive down. The X and Y axes can lie along any two mutually perpendicular directions in the horizontal plane. The points where the anomaly is to be calculated are called field points. These are read from cards. The anomaly for each lamina is calculated in turn and a numerical integration is then performed to obtain the anomaly for the entire body. In addition the potential field and mass per unit area is also calculated.

INPUT:

Cards (via F:1)

Card 1: Sense switch card

Card 2: COORD Initialization card

Transverse Mercator Coordinate for X (FX)

Transverse Mercator Coordinate for Y (FY)

Latitude in Radians for point (X,Y) (RLAT)

Longitude in Radians for point (X,Y) (RLONG)

Starting switch (IST)

G3DCP continued page 2

Input (continued)

Card 3:

Reference Weight (RFW;F8.1)  
Reference Density (RFD;F4.2)  
Reference Gravity (RFG;F5.1)

FILE: (via F:3)

The data representing the body as output by G3DCPREP.

OUTPUT:

PRINTER (via F108)

Printed output of various integration and corrections for each lamina.

USAGE:

In spite of indications to the contrary G3DCP, when used with LSORT and G3DCPREP is reasonably straight forward to use.

STEP 1 Definition of the bodies to be used

Each body for which an anomaly will be calculated must be defined lamina by lamina. A file is created for each body which has a group of records of the following form for each lamina.

Record 1

Lamina number	(NCNT;I2)
Density	(RHO;F10.4)
Vertical distance from origin	(Z;F16.6)

Following Records

X coordinate of polygon point	(X;F12.5)
Y coordinate of polygon point	(Y;F12.5)
Last point flag = 1 for last point in lamina	(LSLPT;I1)

STEP 2 Laminae Sorting

The bodies to be used are input to LSORT. This program will check that a user specified minimum lamina thickness is observed and counts the number of lamina per body.

G3DCP continued page 3

OUTPUT (continued)

### STEP 3

#### Combination of bodies into one model

The bodies to be used in the particular model are combined into one file by program G3DCPREP in a format acceptable to G3DCP. At this point a reference density is specified for each body that will be subtracted from the density in the input file

### STEP 4    Calculation of anomalies

The model and field points are input to G3DCP and the anomalies are calculated.

### SAMPLE RUN

#### STEP 1    Definition of bodies

The bodies must be in the following format whether they are produced by hand or by some modeling program

Body 1                      (file BOD1)

1301

R.No.	DATA
00001	1 1.7700 3.000000
00002	1.7700 3.000000
00003	1.7700 3.000000
00004	1.7700 3.000000
00005	1.7700 3.000000
00006	1.7700 3.000000
00007	1.7700 3.000000
00008	1.7700 3.000000
00009	1.7700 3.000000
00010	1.7700 3.000000
00011	1.7700 3.000000
00012	1.7700 3.000000
00013	1.7700 3.000000
00014	1.7700 3.000000
00015	1.7700 3.000000
00016	1.7700 3.000000
00017	1.7700 3.000000
00018	1.7700 3.000000
00019	1.7700 3.000000
00020	1.7700 3.000000
00021	1.7700 3.000000
00022	1.7700 3.000000
00023	1.7700 3.000000
00024	1.7700 3.000000
00025	1.7700 3.000000
00026	1.7700 3.000000
00027	1.7700 3.000000
00028	1.7700 3.000000
00029	1.7700 3.000000
00030	1.7700 3.000000
00031	1.7700 3.000000
00032	1.7700 3.000000
00033	1.7700 3.000000
00034	1.7700 3.000000
00035	1.7700 3.000000
00036	1.7700 3.000000
00037	1.7700 3.000000
00038	1.7700 3.000000
00039	1.7700 3.000000
00040	1.7700 3.000000
00041	1.7700 3.000000
00042	1.7700 3.000000
00043	1.7700 3.000000
00044	1.7700 3.000000
00045	1.7700 3.000000
00046	1.7700 3.000000
00047	1.7700 3.000000
00048	1.7700 3.000000
00049	1.7700 3.000000
00050	1.7700 3.000000
00051	1.7700 3.000000
00052	1.7700 3.000000
00053	1.7700 3.000000
00054	1.7700 3.000000
00055	1.7700 3.000000
00056	1.7700 3.000000
00057	1.7700 3.000000
00058	1.7700 3.000000
00059	1.7700 3.000000

R.No.	DATA
00060	2 1.7700 3.000000
00061	1.7700 3.000000
00062	1.7700 3.000000
00063	1.7700 3.000000
00064	1.7700 3.000000
00065	1.7700 3.000000
00066	1.7700 3.000000
00067	1.7700 3.000000
00068	1.7700 3.000000
00069	1.7700 3.000000
00070	1.7700 3.000000
00071	1.7700 3.000000
00072	1.7700 3.000000
00073	1.7700 3.000000
00074	1.7700 3.000000
00075	1.7700 3.000000
00076	1.7700 3.000000
00077	1.7700 3.000000
00078	1.7700 3.000000
00079	1.7700 3.000000
00080	1.7700 3.000000
00081	1.7700 3.000000
00082	1.7700 3.000000
00083	1.7700 3.000000
00084	1.7700 3.000000
00085	1.7700 3.000000
00086	1.7700 3.000000
00087	1.7700 3.000000
00088	1.7700 3.000000
00089	1.7700 3.000000
00090	1.7700 3.000000
00091	1.7700 3.000000
00092	1.7700 3.000000
00093	1.7700 3.000000
00094	1.7700 3.000000
00095	1.7700 3.000000
00096	1.7700 3.000000
00097	1.7700 3.000000
00098	1.7700 3.000000
00099	1.7700 3.000000

## BODYZ (FILE BOD2)

1	2.8000	10.00	.00
	39.9998	2000.00000	
	39.9998	-2000.00000	
	-39.9998	-2000.00000	
	-39.9998	2000.00000	
	39.9998	2000.000001	
2	2.8000	10.70	.00
	36.02323	2000.00000	
	36.02323	-2000.00000	
	-36.02323	-2000.00000	
	-36.02323	2000.00000	
	36.02323	2000.000001	
3	2.8000	11.41	.00
	32.05649	2000.00000	
	32.05649	-2000.00000	
	-32.05649	-2000.00000	
	-32.05649	2000.00000	
	32.05649	2000.000001	
4	2.8000	12.11	.00
	25.08473	2000.00000	
	25.08473	-2000.00000	
	-25.08473	-2000.00000	
	-25.08473	2000.00000	
	25.08473	2000.000001	
5	2.8000	12.81	.00
	24.11298	2000.00000	
	24.11298	-2000.00000	
	-24.11298	-2000.00000	
	-24.11298	2000.00000	
	24.11298	2000.000001	
6	2.8000	13.52	.00
	20.14124	2000.00000	
	20.14124	-2000.00000	
	-20.14124	-2000.00000	
	-20.14124	2000.00000	
	20.14124	2000.000001	
7	2.8000	14.22	.00
	16.16948	2000.00000	
	16.16948	-2000.00000	
	-16.16948	-2000.00000	
	-16.16948	2000.00000	
	16.16948	2000.000001	
8	2.8000	14.92	.00
	12.19774	2000.00000	
	12.19774	-2000.00000	
	-12.19774	-2000.00000	
	-12.19774	2000.00000	
	12.19774	2000.000001	
9	2.8000	15.62	.00
	8.22598	2000.00000	
	8.22598	-2000.00000	
	-8.22598	-2000.00000	
	-8.22598	2000.00000	
	8.22598	2000.000001	
10	2.8000	16.33	.00
	4.25426	2000.00000	
	4.25426	-2000.00000	
	-4.25426	-2000.00000	
	-4.25426	2000.00000	

4.25426 2000.000001

STEP 2 Laminae Sorting (see LSORT documentation)

```

!ASSIGN F:1,(FILE,BOD1),(IN)
!ASSIGN F:2,(FILE,BOD2),(IN)
!ASSIGN F:7,(FILE, SORT1),(OUT),(SAVE)
!ASSIGN F:8,(FILE, SORT2),(OUT),(SAVE)
!ASSIGN F:13(FILE, SORTCNT),(OUT),(SAVE)
!LOAD(BI),(UNSAT,(305),(312),(3))
  LSORT Binary Deck
!RUN
!DATA
  0.1
  02
!EOD

```

## STEP 3 Combination of Bodies (see G3DCPREP documentation)

```

!ASSIGN F:1,(FILE, SORT1),(IN),(SAVE)
!ASSIGN F:2,(FILE, SORT2),(IN),(SAVE)
!ASSIGN F:13,(FILE, SORTCNT),(IN),(SAVE)
!ASSIGN F:7,(FILE, G3DCIN),(OUT),(SAVE)
!LOAD (BI),(UNSAT,(3))
  G3DCPREP Binary Deck
!RUN
!DATA
  02
  1.03
  3.3
!EOD

```

FILE G3DCIN

STEP 4      Calculation of Anomalies

```

!ASSIGN F:1,(DEVICE,SI)
!ASSIGN F:3,(FILE,G3DCIN),(IN),(SAVE)
!LOAD (BI),(UNSAT,(514),(456),(305),(312),(3))
!RUN
!DATA
    Sense switch card
    Coord initialization card
    Reference card
    Field point card (s)
                                Last field point card has 1 in col.43
!EOD

```

STORAGE REQUIREMENTS: 30,000 decimal words

SUBPROGRAMS REQUIRED: COORD, STAT, GINOT, PLANET, FORTRAN IV library

PROGRAMMER: Bruce Simon

ORIGINATOR: Carl Bowin

DATE: 1 October 1975

REFERENCES: Bowin, Simon and Wollenhaupt, Mascons a Two Body Solution,  
Journal of Geophysical Research, in press

Talwani, M., and M. Ewing, Rapid Computation of Gravitational  
 Attraction of Three-Dimensional Bodies of Arbitrary Size,  
Geophysics, XXV, 203-225, 1960.

NAME: G3DCPREP

TYPE: Main Program

PURPOSE: To prepare input to program G3DCP

MACHINE: Sigma 7

SOURCE LANGUAGE: Extended Fortran IV

PROGRAM CATEGORY: File management

DESCRIPTION:

G3DCPREP prepares the input to the modeling program  
G3DC from LSORT output.

INPUT:

Cards

Card 1: Number of bodies to be input (NUMBOD;I2)  
Cards 2-6: A reference density for each body to be subtracted  
from the density in the LSORT output (REFD;F10.0)

Files

Files 1-6: 1 file for each of up to 6 bodies to be input  
to G3DC  
File 13: file with laminae count for each body input

OUTPUT:

Files

File 7: a file compatible with G3DC to be used as input  
to G3DC

USAGE: See G3DCP documentation

SUBROUTINES REQUIRED: Fortran IV Library

PROGRAMMER: Lee Gove

ORIGINATOR: Lee Gove

DATE: 1 October 1975

1	2.8000	3.05	.00
	.99998	2000.00000	
	.99998	-2000.00000	
	-.99998	-2000.00000	
	-.99998	2000.00000	
	.99998	2000.000001	
2	2.8000	3.24	.00
	4.89998	2000.00000	
	4.89998	-2000.00000	
	-4.89998	-2000.00000	
	-4.89998	2000.00000	
	4.89998	2000.000001	
3	2.8000	3.44	.00
	8.79997	2000.00000	
	8.79997	-2000.00000	
	-8.79997	-2000.00000	
	-8.79997	2000.00000	
	8.79997	2000.000001	
4	2.8000	3.63	.00
	12.69997	2000.00000	
	12.69997	-2000.00000	
	-12.69997	-2000.00000	
	-12.69997	2000.00000	
	12.69997	2000.000001	
5	2.8000	3.83	.00
	16.59996	2000.00000	
	16.59996	-2000.00000	
	-16.59996	-2000.00000	
	-16.59996	2000.00000	
	16.59996	2000.000001	
6	2.8000	4.02	.00
	20.49995	2000.00000	
	20.49995	-2000.00000	
	-20.49995	-2000.00000	
	-20.49995	2000.00000	
	20.49995	2000.000001	
7	2.8000	4.22	.00
	24.39995	2000.00000	
	24.39995	-2000.00000	
	-24.39995	-2000.00000	
	-24.39995	2000.00000	
	24.39995	2000.000001	
8	2.8000	4.41	.00
	28.29994	2000.00000	
	28.29994	-2000.00000	
	-28.29994	-2000.00000	
	-28.29994	2000.00000	
	28.29994	2000.000001	
9	2.8000	4.61	.00
	32.19994	2000.00000	
	32.19994	-2000.00000	
	-32.19994	-2000.00000	
	-32.19994	2000.00000	
	32.19994	2000.000001	
10	2.8000	4.80	.00
	36.09993	2000.00000	
	36.09993	-2000.00000	
	-36.09993	-2000.00000	
	-36.09993	2000.00000	
	36.09993	2000.000001	
11	2.8000	5.00	.00
	39.99992	2000.00000	
	39.99992	-2000.00000	
	-39.99992	-2000.00000	
	-39.99992	2000.00000	
	39.99992	2000.000001	

NAME: HIG

TYPE: Main Program

PURPOSE: To convert gravity data in the format used by the Hawaii Institute of Geophysics (HIG) to GSUM format.

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:

HIG simply reads a record in H.I.G. format and converts the record to a GSUM record with format forward code of 17

INPUT:

Unblocked tape in HIG format on device no. 1

CARDS:

- 1) Sense switch options - all zero (or blank)
- 2) Source code (I5)
- 3) Input tape serial number (4A1)
- 4) EITP in columns 1 to 4
- 5) Output tape serial number (4A1)
- 6) EOTP in columns 1 to 4

OUTPUT:

Unblocked tape in GSUM format on device no. 2

USAGE:

```
!JOB
!LIMIT
!MESSAGE (I/O mag tape info)
!ASSIGN F:1, (DEVICE, 9T), (SN,xxxx), (IN), (TRIES,10)
!ASSIGN F:2, (DEVICE,9T), (SN,xxxx), (OUT), (TRIES, 10)
!OLAY
!RUN
!DATA
      Data Cards
!EOD
```

HIG continued, page 2

RESTRICTIONS:

SUBROUTINES REQUIRED: GINOT STAT FORTRAV IV library

OPERATIONAL CHARACTERISTICS: Simple read-then-write program

PROGRAMMER: Lee Gove

ORIGINATOR: Carl Bowin

DATE: 1 December 1973

NAME: LSORT  
TYPE: Main Program  
PURPOSE: To edit and sort polygonal laminae of G3DC format  
MACHINE: Sigma-7  
SOURCE LANGUAGE: Extended Fortran IV  
PROGRAM CATEGORY: File Management

DESCRIPTION:

LSORT reads, for each of up to 6 bodies, up to twenty laminae. It counts them, discards laminae of thickness less than ZLIM, and prepares files for input to G3DCPREP

INPUT:

Cards

Card 1: Minimum allowable thickness for a single laminae (ZLIM;F10.0)

Card 2: Number of bodies (files) to be input (NUMBOD;I2)

Files

Files 1-6: 1 file for each of up to 6 bodies

File 13: file of counts of laminae for each body

USAGE: See G3DCP documentation

SUBROUTINES REQUIRED: FORTRAN IV Library

PROGRAMMER: Lee Gove

ORIGINATOR: Lee Gove

DATE: 1 October 1975

NAME: MODPLOT

TYPE: Main Program

PURPOSE: Plots data for preparation of crustal structure models of the earth's crust and plots the output tapes from TALPLOT16

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Graphical display

DESCRIPTION:

This program performs two basic functions. It will plot a profile of data used in the preparation of two-dimensional structure models of the earth's crust as well as plotting the structural models themselves.

Input to this program may include output from a TALPLOT16 run and/or combinations of data output from PROJ4 and/or SAINT2 runs and model polygons. TALPLOT16 is a program that computes the gravitational attraction of two-dimensional structure models. The PROJ4 program projects data from any given area to a selected straight line and outputs a card deck which is then used to plot profiles of the data. The SAINT2 program will interpolate the data output from PROJ4 at regularly spaced intervals (in order to reduce the effects of small local variations) and output data that can also be plotted as profiles.

At the present time, the types of data that the PROJ4 program will process are: GSUM format (containing free-air and Bouguer gravity data, bathymetry and elevation), SPFMT format (containing seismic refraction data), seismicity data, and the model polygon portion of Talplot16 input data.

Program operation is determined by sense switch options selected via the first data card and the order in which the various types of data are arranged in the input deck. The JFMT number (input card 7) informs the program what kind of data follows in the input deck. The data terminator cards (input card 9) indicate completion of input of present JFMT type data. Another JFMT card is then read in. If it equals 9, then the job is terminated.

MODPLOT, page 2

INPUT:

Card 1: Values of SSW(I). Format (80I1) See SSW options under USAGE)

Card 2: XFACT,YFACT,TOP,BOT,BLEFT,RIGHT (6F10.0)

- XFACT - The number of km's/inch in the x-direction (long axis of plot).
- YFACT - The number of km's/inch in the y-direction.
- TOP - The upper bound for the model (km). Elevation above sea level is negative, depth below sea level is positive. To avoid boundary effects, the value of TOP should be more negative than the y-coordinate of the point of highest elevation to be plotted.
- BOT - The lower bound for the model (km). The value of BOT should be a greater positive number than the base of the model, to avoid truncating the bottom of the model.
- BLEFT - The left boundary (km) of the data and model to be plotted.
- RIGHT - The right boundary (km) of the data and model to be plotted.

Card 3: ELFAC,ELDIS,GFAC,GDIS,WFAC,WDIS,PFAC,PDIS (8F10.0)

- ELFAC - The scale factor for elevation (km/in).
- ELDIS - The distance of the origin of the elevation profile above the origin of the model (inches).
- GFAC - The scale factor for gravity; free-air and Bouguer (mgal/in).
- GDIS - The distance of the origin of the plot of free-air and/or Bouguer gravity data above the origin of the structure model (inches).
- WFAC - The scale factor for the model weight profile (kg/in).
- WDIS - The distance of the origin of the weight profile below the base of the model.
- PFAC - The scale factor for individual polygon contributions (mg/in).
- PDIS - Distance of the origin of the curves (contribution of individual polygons) above the origin of the structure model.

## MODPLOT, page 3

Card 4: HT, DBOT (2F10.0)

- HT - Character height multiplication factor in multiples of 0.07" (used in the call to symbol for the plotting of the anomaly curves). If HT = 0 a default value of 3 is assumed. (0.21").
- DBOT - The distance that the weight curve is supposed to be plotted above the bottom of the plot.

Omit the following cards when plotting only the output from TALPLOT16.

Card 5: A label card containing 80 columns of alphanumeric data. This will appear at the beginning of the plot. Include this label card only if SSW(1) = 1. If SSW(1) = 0 or 2, then the label is obtained from the TALPLOT16 output tape.

Card 6: Crustal structure section (CSS) identification card (same card as used in the PROJ4 run).  
CSS ID, ANG, DMAXM, ILAT, RILTM, ILONG, RILDm, LABEL  
(8A1,2X,2F10.0,I4,F6.2,I4,F6.2,30A1)

- CSS ID - Crustal structure section identification number (CSS-NNN).
- ANG - The angle between the vertical and the straight line to which the data has been projected in the PROJ4 program.
- DMAXM - The greatest distance (km) from the reference point for which data will be accepted.
- ILAT - The latitude, in degrees, of the reference point for the line.
- RILTM - The latitude, in decimal degrees, of the reference point.
- ILONG - The longitude, in degrees, of the reference point.
- RILDm - The longitude, in decimal degrees, of the reference point.
- LABEL - A label containing up to 30 alpha-numeric characters

Card 7: JFMT (J-format) (I1)

<u>JFMT</u>	<u>For</u>	<u>JFMT Data Terminator (I1)</u>
1 (or 2)	GSUM	2 - "8" cards (column 1)
3	SPFMT	2 - "8" cards
4	SEISMICITY	1 - "8" card
5	MODEL POLYGONS	none
6	TALPLOT16 INPUT	none

Note: 5 and 6 both read TALPLOT16 input. JFMT=6 reads from card 1 of TALPLOT16. JFMT=5 reads from after card 5 of TALPLOT16. In either case, only the model polygons will be plotted.

A. If input is on cards:

Card 8: data cards for the JFMT

Card 9: data terminator cards (see card 7)

B. If input created by the PROJ4 program is on mag tape, and JFMT = 1 or 2 for GSUM format:

Card 8: input tape serial number (I4)

Card 9: EITP in columns 1-4.

C. If input created by the PROJ4 program is on tape, but is some format other than GSUM, there will be no cards 8 or 9. Data termination cards are already on the mag tape.

Repeat cards 7 - 9 for each JFMT data type to be plotted.

Card 10: JFMT = 9 to terminate the job (I1)

OUTPUT

- 1) on line printer: the input parameters
- 2) on Calcomp plotter: profiles of the data and/or two-dimensional plot model input to the program.

USAGE:

## Sense Switch Settings

- SSW(1) = 0 to plot only from TALPLOT16 output tape  
           = 1 to plot PROJ4 data, and/or model polygons: reads JFMT.  
           = 2 to plot both TALPLOT16 output tape and PROJ4 data
- SSW(2) = 1 to plot Bouguer anomaly in addition to the free-air
- SSW(3) = 1 to plot height from GSUM data
- SSW(4) = 1 to plot elevation. The input values are on the TALPLOT16 output tape (in meters). Use only if elevation data was input to previous TALPLOT16 run.

SSW(5) = 1 to plot observed and calculated gravity.  
 SSW(6) = 1 to plot weight  
 SSW(7) = 1 to plot contribution of each polygon.  
 SSW(8) = 1 to plot two-dimensional Bouguer anomaly  
 in TALPLOT16 output.  
 SSW(13) = 1 to print intermediate values.  
 SSW(14) = 1 to plot only a dot for GSUM free-air  
 values, rather than a continuous line.  
 Be sure that SSW(14) = 1 if plotting from unsorted  
 PROJ4 output.  
 SSW(30) = 1 to read GSUM data on 2 cards  
 SSW(32) = 1 to read SPFMT data on 2 cards  
 SSW(36) = 1 to read Seismicity data from cards  
 SSW(44) = 1 to read Seismicity data in blocked format  
 The calculated anomaly plot is marked by asterisks.

#### RESTRICTIONS:

Elevation above sea level is negative, depth below sea  
 level is positive.

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: ANOV3, DISAZ, DMTOR, DNAV, EVIL, EXIT, EXTD,  
 GINOT, INCEP, ISW, PINOT, PLOT, PLOTS, SPLOT, SPOT, STAT,  
 SYMBOL, WHERE, YINOT, STANDARD FORTRAN IV LIBRARY.

TIMING: TALPLOT16 output of approximately 200 polygon points  
 will take just under two minutes of computer time.  
 For other data types, the timing is undetermined.

#### ERRORS AND DIAGNOSTICS:

<u>Message</u>	<u>Error</u>
"Plot too wide, width = VIT"	Distance of model origin above the bottom (right) of paper is too great. The value of VIT must be less than 29. (VIT = DBOT + WDIS + YWIDE/ YFACT) where (YWIDE = TOP-BOT)

PROGRAMMER: Carl O. Bowin

ORIGINATOR: Carl O. Bowin

DATE: 12 June 1974

NAME: NOAA  
TYPE: Main Program  
PURPOSE: To convert gravity data in NOAA format to  
 WHOI GSUM format  
MACHINE: Sigma-7  
SOURCE LANGUAGE: Fortran IV  
PROGRAM CATEGORY: Format Conversion

DESCRIPTION:

NOAA uses asynchronous I/O to speed up the conversion of NOAA data to GSUM. The GSUM record has a format forward code of 16. The input tape is blocked by 20 and the output is blocked by 50. Input gravity data is assumed to be referenced to International Gravity Formula 1930. Conversion to IGF 1967 is made in another program.

INPUT:

- 1) Mag-tape in NOAA format blocked by 20
- 2) Cards
  - 1) Source Code(I5)

OUTPUT:

- 1) Mag tape in WHOI GSUM format blocked by 50

USAGE:

```

!JOB
!MESSAGE (Mag tape info)
!ASSIGN F:1, (DEVICE,9T), (SN,xxxx), (IN), (TRIES,10)
!ASSIGN F:2, (DEVICE,9T), (SN,xxxx), (OUT), (TRIES,10)
!OLAY
!RUN
!DATA
    Data card
!EOD
  
```

RESTRICTIONS:

- a) NOAA tape must be blocked by 20
- b) GSUM tape blocked by 50 with format forward code = 16

NAME: PROFG

TYPE: Main Program

PURPOSE: Profiles GSUM data

MACHINE: XDS Sigma 7

SOURCE LANGUAGE Fortran IV

PROGRAM CATEGORY Graphical Display

DESCRIPTION:

This program profiles various parameters in GSUM format. By using GINOT as the input routine all the options in GINOT are available.

INPUT:

- 1) GSUM data (via F:1)
  - GSUM data on tape or cards, blocked or unblocked
- 2) Cards:
  - Card 1 Sense switch card
    - SSW(5) = 0 no effect
      - = 1 to select station
    - SSW(14)= 0 no effect
      - = 1 process only if data after start date
    - SSW(25)= 0 no effect
      - = 1 to call MOUNT
    - SSW(29)= 0 no effect
      - = 1 process only specified source codes
      - = 2 skip specified source codes
      - = 3 process only specified format-forward codes
      - = 4 skip specified format-forward codes
    - SSW(30)= 0 no effect
      - = 1 input GSUM on cards
    - SSW(40)= 0 input unblocked with no DLT
      - = 1 input blocked X50 with no DLT
      - = 2 input blocked X50 with DLT
    - SSW(41)= 0 output on tape unblocked
      - = 1 output on tape blocked x 50
    - SSW(46)= 0 no effect
      - = 1 process only within specified geographic bounds
    - SSW(47)= 0 no effect
      - = 1 process only within specified time interval
      - = 2 skip data within specified time interval

PROFG continued

INPUT continued

- Card 2 GINOT initialization card
- Card 3 Number of nautical miles, kilometers, or hours per inch  
on plot (DIFAC;F10.0)  
Engineering units per inch in Y direction (YFAC;F10.0)  
Number of points between time annotation (LCNT;I5)  
0 For nautical miles, 1 for kilometers (MIKM;I5)  
Number of plots (NPLOT;F5)  
Number of files (NFILE;I5)
- Card 4 Upper limit for plotting data value in eng units (ULIM;F10.0)  
Lower limit for plotting data value in eng units (BLIM;F10.0)  
Distance limit in inches for plotting data (DLIM;F10.0)  
1 for distance along track 0 for time along track  
(IXDIR; I5)
- Card 5 Allowable distance for length of plot before reinitialization  
(XALOW; F10.0)  
Inches to move before reinitialization (DMOVE;F10.0)
- Card 6 Starting day for processing (LIMDA; I5)  
Starting month for processing (LIMMO; I5)  
Starting year for processing (LIMYR; I5)  
Starting hour and minute for processing (LIMHM; I5)
- Card 7 Station number to be selected (MSTA; I5)

OUTPUT:

- 1) Printer (via F:108)  
All input parameters are output to the line printer along  
with error messages
- 2) Plot tape (via F:PLOT)  
A plot tape for plotting by the 30" Calcomp plotter

USAGE:

```
!JOB
!LIMIT (9T,1), (7T,1), (CORE,20), (TIME,X)
!MESSAGE GSUM data tape info
!MESSAGE plot tape info
!ASSIGN F:1 to GSUM file
!ASSIGN F:PLOT, (DEVICE,7T), (SN,PLT1), (BIN), (UNPACK), (TRIES,10)
!LOAD (EF, (PROFG,456), (STAT,3)), (UNSAT, (305), (312), (3))
!RUN
!DATA
!EOD
```

RESTRICTIONS: None

PROFG continued

STORAGE REQUIREMENTS: 15K

SUBPROGRAMS REQUIRED: GINOT, FIND, ISW, CDATE, MCVOL, SPOT

TIMING: Undetermined

ERRORS AND DIAGNOSTICS

DLIM (date) distance limit exceeded  
ULIM (date) upper limit exceeded  
BLIM (date) bottom limit exceeded  
END DATE PASSED end of processing

PROGRAMMER: Carl Bowin and Lee Gove

ORIGINATOR: Carl Bowin

DATE: 2 October 1975

NAME: PROJ4

TYPE: Main Program

PURPOSE: To project data onto a given line providing the data is within a given area and within a given distance from the line.

MACHINE: Sigma-7

SOURCE LANGUAGE: FORTRAN IV

PROGRAM CATEGORY: Data analysis

DESCRIPTION: This program projects several data formats to a straight line defined by input latitude and longitude of a point, and bearing from the north (0 to 360 degrees). The program can process several different formats in the same run, the order for processing being determined by a format choice input card (card 6).

The data can be output to a tape to be used in MODPLOT, TALPLOT16 and/or SAINT 2 runs when JFMT = 1 or 2 (GSUM format).

The different data formats that the program will process are:

GSUM - Gravity summary format, containing free-air and Bouguer gravity values, elevation and bathymetry.

SEAG1 or 2 - Sea gravity data format, containing corrected gravity values plus magnetics. SEAG data may be input in 1939 IGF and referenced to Potsdam (SEAG1), or in 1967 IGF and referenced to the IGSN71 (SEAG2). GSUM data may be input in 1939 IGF and referenced to Potsdam (IREC=1), or in 1967 IGF and referenced to the IGSN71 (IREC=2). For either SEAG or GSUM input, output is GSUM with IREC=2.

SPFMT - Seismic refraction data format.

SEISMICITY - U.S. Coast and Geodetic Survey data format.

ACTIVE VOLCANOES - File on active volcanoes key-punched from "Catalogue of the Active Volcanoes of the World" and supplementary references.

-2-

INPUT:

Card 1: Crustal structure section (CSS) identification card containing projection parameters.

CSS ID, ANG, DMAXM, ILAT, RILTM, ILONG, RILDm, LABEL  
(10A1, 2F10.0, I4, F6.2, I4, F6.2, 30A1)

- CSS ID - Crustal structure section identification number (CSS-NNN).
- ANG - The angle between the vertical and the straight line to which the data will be projected in the PROJ4 program (0 to 360 degrees).
- DMAXM - The greatest distance in nautical miles from the reference point for which data will be accepted.
- ILAT - The latitude, in degrees, of the reference point for the line. South is negative.
- RILTM - The latitude, in decimal minutes, of the reference point. Note: if degrees are negative minutes must be negative also.
- ILONG - The longitude, in degrees, of the reference point. West is negative.
- RILDm - The longitude, in decimal minutes, of the reference point. Note: if degrees are negative minutes must be negative also.
- LABEL - A label containing up to 30 alpha-numeric characters.

Cards 2-5: Four boundary cards (I4, F6.2)

- |    |         |         |                   |
|----|---------|---------|-------------------|
| 2) | Degrees | Minutes | (top latitude)    |
| 3) | "       | "       | (bottom latitude) |
| 4) | "       | "       | (left longitude)  |
| 5) | "       | "       | (right longitude) |

Card 6: Selection of order that data formats are processed:  
(9I5) 1st JFMT, 2nd JFMT, 3rd JFMT, ..., 9th JFMT.

JFMT = 1 for GSUM format.

JFMT = 2 for SEAG1 or 2 input, output is GSUM format with IREC=2.

JFMT = 3 for SPFMT format.

JFMT = 4 for SEISMICITY data format.

JFMT = 7 for Active Volcanoes.

-3-

Card 7 and following vary depending on the JFMT. Repeat cards 7 and greater for each selected format in the order given on card 6. The program terminates when JFMT = 0 or blank, or when start date = 99.

A. for JFMT = 1      GSUM format      (uses subroutine GINOT)

Card 7: Sense switch options. Put sense switch (0) option in column 80, all others in column corresponding to switch number.

ISW (3) = 1 to output GSUM data onto a disk file (unit 20).

ISW (4) = 1 to output intermediate values for checking (subroutine PROJ).

ISW (12) = 1 to list date identification.

ISW (26) = 1 to output on high speed printer only.

ISW (27) = 1 to suppress rewind of ITAPE at start of job.

ISW (27) = 2 to suppress rewind of JTAPE at start of job.

          = 9 to suppress rewind of both ITAPE and JTAPE.

ISW (29) = 1 to read and test for selected source code numbers to be processed.

          = 2 to read and test for selected source code numbers to be skipped.

ISW (30) = 1 for input data on cards.

ISW (31) = 1 to output data on cards.

ISW (40) = 0 to process without the DLT.

          = 1 to process with bounds using the data location table (DLT).

ISW (60) = 1 to process only data with IFFC = 4, ABSTRACTOR output.

ISW (61) = 1 to replace free-air, Bouguer, elevation, latitude and longitude with averaged values.

Card 8: (optional) If ISW (29) = 1 or 2, enter ISRC numbers to be processed or skipped (1615).

Card 9: (optional) If data is input on tape, serial number of input tapes - one per card (A4). If data is not input on tape, ISW (30) must equal 1.

-4-

Card 10: (optional) If there are card(s) 9, EITP in columns 1-4.

Card 11: (optional) If data is to be output to mag tapes, serial number of output magnetic tapes, one per card (A4). If no output tapes, then no serial number cards and either ISW (31) or ISW (3) must equal 1.

Card 12: (optional) If there are card(s) 11, EOTP in columns 1-4. If no output tapes, then no Card 12.

Card 13: Start and end dates, ISKP card  
 ISTDA, ISTMO, ISTYR, ISTHM, IENDA, IENMO, IENYR, IENHM,  
 ISKP (3I2, I4, 5X, 3I2, I4, 5X, 3I5)  
 Start date for processing  
 ISTDA (day)  
 ISTMO (month)  
 ISTYR (year)  
 ISTHM (time of day)  
 End date for processing  
 IENDA (day)  
 IENMO (month)  
 IENYR (year)  
 IENHM (time of day)

ISKP Number of records to be skipped at the start of the job, is much faster than using start date only.

Use a blank card if start and end dates not used. If this card is blank, plotting begins with the first record.

B. for JFMT = 2 SEAG1 or 2, format input, GSUM output with IREC=2 (uses SINOT and GINOT) subroutines)

Card 7: Sense switch options. Same as for JFMT = 1.

C. for JFMT = 3 SPFMT format (uses subroutine PINOT)

Card 7: Sense switch options.  
 ISW (4) = 1 to output intermediate values for checking (subroutine PROJ).  
 ISW (26) = 1 to output to line printer.  
 ISW (32) = 1 to read SPFMT data on two cards.  
 ISW (33) = 1 to write SPFMT data on two cards.

Card 8: Start and end dates and ISKP. (Same as Card 13 on JFMT = 1 above).

D. for JFMT = 4 SEISMICITY format (uses subroutine YINOT)

Card 7: Sense switch options.

ISW (4) = 1 to output intermediate values for checking (subroutine pros).

ISW (26) = 1 to output to line printer.

ISW (44) = 1 to read blocked input.

ISW (45) = 1 to write blocked output

Card 8: Start and end dates and ISKP (Same as card 13 on JFMT = 1 above).

E. for JFMT = 7

at present there is no volcano routine.

#### OUTPUT:

On unit 2:

- 1) JFMT number record (11, 20X).
- 2) projected data at appropriate format.
- 3) two records with a numeral '8' in column 1 (11, 20x). Only one '8' record for seismicity and volcano data.

This sequence is repeated for each selected JFMT.

On unit 20 (disk): GSUM data for subsequent processing by SAINT 2 if SSW(3) = 1 (Only applies when JFMT = 1 or 2).

On unit 108 (line printer): initialization parameters.

USAGE: Assign cards vary depending on format. There should be a separate assign card for each JFMT input.

Unit 2 is for output of projected data

Unit 3 is for GSUM format input

Unit 4 is for SEAG1 format input; output is GSUM with IREC=2

Unit 5 is for SPFMT format input

Unit 6 is for SEISMICITY format input

Unit 9 is for Active Volcano data input

Unit 20 is for disk storage of GSUM data if ISW(3) = 1.

To stop processing make start day = 99. If start day = 99, the program goes to 1000 and checks input format choices for format code number of zero.

-6-

RESTRICTIONS: South and west are negative.  
Degrees and minutes must have the same sign  
(e.g. 47°45.0' or -32° - 30.0').

STORAGE REQUIREMENTS: Unknown.

SUBPROGRAMS REQUIRED: AREAK, DNAV, ENDIO, ENDLT, EVIL, EXIT, FIND, GINOT,  
ISW, MCVOL, MOUNT, NAVIN, NAVOT, OBG, PINOT, PKBY,  
PROJ, SETSKP, SINOT, SKPREC, STAT, TODAY, UNPKBY,  
YBLKI, YBLKO, YINOT.

TIMING: Undetermined.

ERRORS AND DIAGNOSTICS:

MESSAGE

Error in SKPREC, IND = 'IND'

ERROR

number of records to be  
skipped = 0.

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 24 February 1975

NAME: RETRIEVE

TYPE: XDS Manage Processor Program

PURPOSE: To extract data from a data base file according to user specified search criteria.

DESCRIPTION:

Retrieves data from the data files using a selective criterion applied to any part of the data record as described in the dictionary. This selection may involve a secondary file called "matchine file" to which the main data file is compared. The Manage Reference Manual gives detailed information on the use of the program.

USE:

There is a tape containing information on world seismicity (hypocenter data) from 1900 through October 1974, which is available for general use. The tape will be updated periodically, and will always be assigned the same number. The serial number is #L20.

Because of the amount of data on this tape (about 90,000 earthquakes), we recommend you retrieve the data in the area you wish to plot, using the RETRIEVE processor, before charting. A sample RETRIEVE run is included here. Input bounds are in order TOP, BOTTOM, LEFT, RIGHT. South and west are negative. For more details see the XEROX MANAGE processor manual. If a listing of the data retrieved is desired, it can be dumped using FTDUMP. For details see the Handbook for Computer Users.

The program CHART can give you plots annotated with date, depth in km, or magnitude; or as in the sample run, it can plot symbols whose type and size varies with depth and magnitude. For additional details, see the DDL Documentation for CHART.

AD-A035 454

WHOI-77-2  
GRAVITY DATA PROCESSING PROGRAMS.(U)

FEB 77

CARL BOWIN

UNCLASSIFIED WOODS HOLE OCEANOGRAPHIC INSTITUTION, MA

2 OF 6  
AD-A  
035 454



# Office Memorandum • WOODS HOLE OCEANOGRAPHIC INSTITUTION

TO : Distribution

DATE: 5 September 1973

FROM : Carolyn Dean

SUBJECT: Program RETRIEVE

Because of the way this program 'finds' data; what you see is not what you get.

For example, suppose you are trying to retrieve all data within the bounds 47, 45, -118, -116: that is, all data in squares marked X.

					49
					48
	Y <sub>1</sub>	Y	Y		47
	X	X	Y		46
	X	X	Y		45
					44
-119	-118	-117	-116	-115	-114

The retrieve program, however, uses your input bounds to calculate keys to match against a key in the data on the input tape. The sort keys are created by adding 90 to the latitude and 180 to the longitude, in order to insure positive numbers for the codes for all points of the earth. The sort key for latitude 47 would be 137, but this code refers to data in the degree square to the north of the given latitude line. Similarly, the sort key for longitude -118 would be 62, but this number refers to the area of the degree square to the east of the given longitude line. Thus, in the example given, the code 137062 refers to the square marked Y<sub>1</sub>.

If your input bounds are 47, 45, -118, -116, the data that will find its way to your output is all data in squares marked X and all data in squares marked Y.

Therefore, you must be aware of this quirk and learn to live with it, or you must hedge on your input bounds. To get only that data in squares marked X, your input bounds would be 46, 45, -118, -117.



NAME: SAINT2

TYPE: Main Program

PURPOSE: To interpolate data at even intervals

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Data Analysis

DESCRIPTION:

This program operates on GSUM format output of PROJ4, a projection program. SAINT2 will sort the data by distance along the projected line, if necessary, and then interpolate the values at even intervals of distance, if desired. Interpolation is accomplished by filtering with a polynomial function, certain parameters of which are entered at run time.

The phrase "interpolated GSUM from SAINT2" is entered in the free-field portion of the GSUM output record.

INPUT:

Card 1: Sense switch values (put a "1" in column corresponding to switch number of desired option)

ISW(1) = 1 to sort data  
ISW(2) = 1 to punch sorted data  
ISW(3) = 1 to print sorted data  
ISW(4) = 1 to interpolate data  
ISW(5) = 1 to output interpolated data in GSUM format  
(see ISW(11))  
ISW(6) = 1 to punch interpolated free-air anomaly in format  
for TALPLOT  
ISW(7) = 1 to print interpolated data and summary of job.  
ISW(8) = 1 to print input data  
ISW(9) = 1 to punch interpolated elevation data in format  
for TALPLOT  
ISW(10) = 1 to read GSUM data from magnetic tape  
ISW(11) = 0 to punch interpolated GSUM on cards (two cards  
per record)  
= 1 to write interpolated GSUM on magnetic tape

Card 2: XKM,XSCAL,NPTS (F5.1, F5.1, I5)

XKM = distance between interpolated points in kilometers.  
XSCAL = maximum distance in kilometers for which points  
are given unit weight in interpolation algorithm.  
Beyond this weight drops as  $X/XSCAL$ . Good first  
approximation:  $XSCAL = 3 * XKM$ .  
NPTS = number of points on either side of output point  
to be used in interpolation filtering function.  
Default value is 4.

Card 3: Crustal structure section (CSS) identification card  
(same card as used in PROJ4 run).

CSS ID,ANG, DMAXM, ILAT, RILTM, ILCNG, RILDm, LABEL  
(10A1, 2F10.0, I4, F6.2, I4, F6.2, 30A1)

CSS ID = crustal structure section identification number  
ANG = the angle between the vertical and the straight  
line to which the data has been projected.  
DMAXM = the greatest distance from the reference point  
for which data will be accepted, in nautical miles.  
ILAT = the latitude, in degrees, of the reference point  
for the line. North is positive.  
RILTM = the latitude, in decimal minutes, of the reference  
point (note: if degrees are negative, minutes  
must be negative, too).  
ILONG = the longitude, in degrees, of the reference point.  
West is negative.  
RILDm = the longitude, in decimal minutes, of the reference  
point (see note Re: RILTM)  
LABEL = a label containing up to 30 alpha-numeric characters.

Card(s) 4: If ISW(10) is not set, GSUM format input data  
cards (output of PROJ4).

Card 5: If data is on cards, a terminator is needed. Either  
a !EOD card or two cards with a '9' in column 1.

OUTPUT:

On line printer: input parameters and listings controlled by sense switches.

On card punch: sorted data, GSUM format, TALPLOT elevation input, and/or TALPLOT free-air anomaly input, as requested.

On magnetic tape: Interpolated GSUM format, if desired.

USAGE:

Assign F:1 to input device (either magnetic tape or card reader)

Assign F:2 to final output tape.

Assign F:3 to RAD output file for sort

Assign F:4 to RAD input file for sort

(These must have the same file name. SAINT2 writes the file and then sorts it.)

Assign F:5 to RAD output file for output of sort. (OUT) should be on the assign card. It will be changed to (IN) by the sorter, so that SAINT2 can read it in.

Here are sample assign cards:

```
ASSIGN F:1,(DEVICE,SI)
ASSIGN F:2,(DEVICE,9T),(OUT),(SN,GN30),(TRIES,10)
ASSIGN F:3,(FILE,FILE1),(OUT),(SAVE)
ASSIGN F:4,(FILE,FILE1),(IN),(SAVE)
ASSIGN F:5,(FILE,FILE3),(OUT),(SAVE)
```

RESTRICTIONS: Maximum number of input records read is 999. The number of interpolated records will be  $\leq 1000$ .

STORAGE REQUIREMENTS: 1575 decimal words. Requires (CORE,16) on limit card.

SUBPROGRAMS REQUIRED: AMAW, AMIW, ATSM, CLOFIL, DISAZ, DMTOR, EQN, OPIN, PICK, SETAL, THORT, WT, WTSET

TIMING: less than 3 minutes for under 150 records input, 200 records output, with two listing options selected.

ERRORS AND DIAGNOSTICS: NONE

PROGRAMMERS: Jack Wolfe, Carl Bowin, Allin Folinsbee

ORIGINATOR: Carl Bowin

DATE: 2 October 1974

DESCRIPTION: Program CRWT3 (which see) outputs University of Toronto World Seismic Refraction Compilation in W.H.O.I. SPFMT format. SELSP will select from among this output those records satisfying the required criteria, such as quality, geographic area, geologic province, depth to mantle, or water depth. Records which satisfy the criteria specified on the input cards are output in SPFMT format. A record key of 9 causes that record to be skipped.

INPUT:

Card 1: Sense switch options (80 Il) Put sense switch 0 option in column 80, others in column corresponding to switch number.

```

ISW(26)  = 1,  to output on line printer only (sub. PINOT)
ISW(32)  = 1,  to read data from two cards per record
            (sub. PINOT)
ISW(33)  = 1,  to write data onto two cards per record
            (sub. PINOT)

```

Card 2: (6I5, 2F 10.0)

```

JMET  = 1  to sort on the basis of quality
IMET  degree of quality required
      = 1  for reliable interpretation
      = 2  for less-reliable interpretation
      = 3  for unreliable interpretation
JPROV = 1  to sort on the basis of province
IPROV  is the province number to be selected
        (see references)
IAR    = 1  to sort on the basis of area bounds
IDMP   = 1  to sort on the basis of mantle depth
SMIND  Minimum depth to mantle in kilometers
SMAXD  Maximum depth to mantle in kilometers

```

## INPUT (continued)

Card 3: (I5, 2F10.0)

IDP = 1 to sort on the basis of water depth  
      =-1 to sort on basis of elevation above sea level  
          (land data)

DMIN minimum water depth or elevation in kilometers  
DMAX maximum water depth or elevation in kilometers  
(All positive values)

Cards 4-7 (optional)

Area bounds (I5, F10.0)

If IAR=1, enter top bound, bottom, left  
and right, in integer degrees, and minutes  
with decimal point, one bound per card.

Data cards (optional):

If sense switch 32 is set to one, SPFMT cards should  
follow, two cards per record.

OUTPUT:

To line printer: input parameters and counters of  
records read and written

If ISW(26)= 1, selected output records

On output device: records which satisfy selection criteria,  
which may be output on cards or magnetic tape.

USAGE:

Assign F:1 to input device. Unless ISW(26) or ISW(33) = 1,  
assign F:2 to an output tape.

RESTRICTIONS:

STORAGE REQUIREMENTS: 468 decimal words; (CORE,9) is required on  
LIMIT card.

SUBPROGRAMS REQUIRED: ARCK, ARLIM, EXIT, PINOT

TIMING: 5 charge units to select 100 records from 2300 (output to  
lister).

ERRORS AND DIAGNOSTICS: None

SELSP (continued)

- 3 -

November 21, 1975

REFERENCES:

McConnell, R.K., and G.H. McTaggart - Cowan, Crustal  
Seismic Refraction Profiles, A Compilation, 1963.  
Univ. of Toronto, Sci. Rept. No. 8, 54 pp.

PROGRAMMER: CARL BOWINORIGINATOR: CARL BOWINDATE: 6 August 1975

NAME: SPFMT

TYPE: Main Program

PURPOSE: Converts seismic refraction data in University of Toronto World Seismic Refraction Compilation format to W.H.O.I. SPFMT format

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:

SPFMT converts seismic refraction column data in University of Toronto World Seismic Refraction Compilation format (one or two cards per profile) to W.H.O.I. SPFMT format (each 120 character output record is one profile). Input data is assumed to be on cards.

INPUT:

card 1: Sense switches (blank card)

No sense switches have been implemented for this program, available for possible future use.

card 2: data cards

card 3 blank card to signal end of input data. without this card, the last input card may not be output.

OUTPUT: On device no. 2 - seismic refraction column data in SPFMT format.

USAGE: Assign F:1 to input device; F:2 to output device

RESTRICTIONS: NONE

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: EVIL, ISW, STAT

TIMING: Not determined

SPFMT

ERRORS AND DIAGNOSTICS: 'EDF FOUND, NREC =  
NREC is the number of records output

PROGRAMMER: John Woodside, Carl Bowin

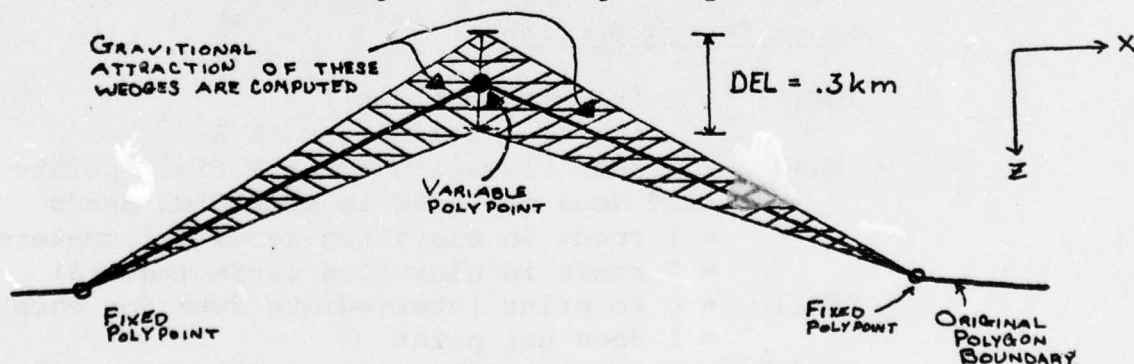
ORIGINATOR: Carl Bowin

DATE: 6 August 1975

NAME: TALPLOT 16  
TYPE: Main program  
PURPOSE: To compute gravity anomalies for a set of two-dimensional polygons  
MACHINE: Sigma-7  
SOURCE LANGUAGE: Fortran IV  
PROGRAM CATEGORY: Data analysis  
DESCRIPTION:

This program uses Talwani's method for calculating the gravity anomaly for two-dimensional polygons. It sums the contributions from the set of polygons and compares the calculated result to the observed gravity which is read in at the start of the program. The calculated gravity is referenced to the observed gravity at a specified point by subtracting the value of ( $g_{\text{calculated}} - g_{\text{observed}}$ ) at specified point from the calculated values at each point.

The program calculates the RMS difference between the calculated and observed gravity anomaly. If desired the program will modify specified points of the last polygon that is read in so as to reduce the RMS difference. This is done in the following way. For each point in the polygon that is to be modified, the effect of a change of  $z$  for this point on the gravity observed at each field point is calculated. This is done by computing the effect of thin triangles on the gravity at the field points.



If we have  $N_{\text{FIELD}}$  field points and  $N_{\text{VAR}}$  variable polypoints (note that  $N_{\text{VAR}} \leq N_{\text{FIELD}}$ ) we end up with a set of over-determined equations for the change in depth of each of the variable polypoints. These  $N_{\text{FIELD}}$  equations are reduced to  $N_{\text{VAR}}$  equations by the standard summation method.

TALPLOT 16 continued, page 2

# DESCRIPTION (continued)

The equations are solved, and the appropriate modifications are made to each of the polypoints. The new gravity contribution of the polygon is calculated, along with the RMS error. We then go back and calculate a new set of variational parameters, and then another polygon model, and so on. This loop continues until one or the other of the following two things happens: 1. the number of new polygons calculated exceeds IMAX or, 2. the RMS error decreases by less than 0.5 mgals. When this happens the program prints out the new final polygon, prints out some graphs of the observed and calculated gravity, and then stops. It also calculates the crustal weight for a 1 cm<sup>2</sup> column down to the base of the model. If the new polygon displaces more than one polygon, above or below, the gravity and mass calculation will be incorrect.

## INPUT:

### First Card:

A card containing 80 columns of alphanumeric data. This will appear at the start of the job and at start of plot. Format (20A4)

### Second Card: Sense Switches (80I1)

Put ISW(0) option in column 80 and the other options in columns corresponding to the switch number.

#### Sense Switch Settings

ISW(1) = 0 to write onto unit 2  
           = 1 not to write onto unit 2  
 ISW(2) = 0 sets elevation for all field points = 0  
           and does not read in elevation cards  
           = 1 reads in elevation cards (kilometers)  
           = 2 reads in elevation cards (meters)  
 ISW(3) = 0 to print intermediate data for each polygon  
           = 1 does not print  
 ISW(4) = 0 to plot intermediate data for each polygon  
           = 1 does not plot  
 ISW(5) = 0 not to plot of elevation on final plot  
           = 1 to plot elevation on final plot  
 ISW(6) = 1 to adjust last polygon to fit gravity data  
 ISW(7) = 1 to print intermediate data for the fitting  
           part of the program (debugging only)

## TALPLOT 16 continued, page 3

## INPUT (continued)

- ISW(8) = 1 to disable the punching of new polygon points
- ISW(9) = 1 NOT to write intermediate data for each polygon onto unit 2 (JTAPE)
- ISW(12)= 1 to correct calculated weights with an elevation factor of  $(\text{elev}(\text{km}) * 2.67 * 100)$ . Use when model does not contain that part of topography above sea level.
- ISW(13)=1 to print out results of input polygon before varying the boundary

Third Card: RDENS, RWGT, RHOD, REFX, FXI, DELFX, M, NFER, IMAX  
(6F10.2, I10, 2I5)

- RDENS - The reference density that is subtracted from the density of the polygons, (usually 0.0).
- RWGT - The weight that is to be subtracted from the crustal weight calculated by the program. The value should be the normal weight of a normal column of material 1  $\text{cm}^2$  down to the bottom of the model in kg.
- RHOD - The difference in density that is used by the model adjusting program. It is equal to the density of the layer below the adjustable boundary minus the density of the layer above the adjustable boundary.
- REFX - The value of the X coordinate at which the calculated gravity is set equal to the observed gravity.
- NFER - half length of the filter used to smooth the residuals before modifying the lower polygon boundary. If = 0 no smoothing is done.
- FXI - The X coordinate for the first field point.
- DELFX - Distance in Km between field points
- M - The number of field points (must be .LE. to 200)
- IMAX - The maximum number of times the boundaries of the last polygon will be adjusted (5 is a good number).

## FOURTH Card: Observed gravity

The observed gravity values, five per card (5F10.1) (a total of M values). If the observed gravity is not known the value of the observed gravity should set to 990.

## Fifth Card: (Optional: if ISW(2)=1) elevations

Elevation of the field points in km, five per card(5F10.1)  
(a total of M values)

## TALPLOT 16 continued, page 4

Then comes a group of cards that is repeated for each polygon

repeated for each polygon

LNO, RHORK (I5,F10.3)

XX,ZZ,ICODE,IAL (2F10.2, 2I1) a card like this for each polypoint.

- LNO - The number of the polygon  
 = 1 this is a water layer, and the program will calculate the 2-D Bouguer anomaly  
 = 99 this is the last polygon in the model  
 = any other number for ordinary polygon
- RHORK - The density of the polygon (in gm/cc)
- XX,ZZ - The X and Z coordinates of the polygon points.  
 Note that Z is positive down and X is positive to the right (the coordinates are in km)
- ICODE = 9 to indicate that this is the last point in the polygon.  
 = blank if not the last point.  
 Note that the last polygon point must have the same coordinates as the first polygon point, and that the polygon should be given in a clockwise order.
- IAL - Used only by the boundary altering part of the program.  
 = 1 to indicate that this polygon point is to be varied. For stability the number of points to be varied should be less than  $1/4$  the number of observed gravity values, although the number of points can be equal to the number of observed gravity values if a perfect fit is desired. Also, all the points in a boundary should not be varied as this will lead to instability in the model. The final restriction is that the first or last point in a polygon must not be varied. If you wish to vary these points the polygon points should be reordered.

TALPLOT 16 continued Page 5

OUTPUT:

Onto URN 2 if ISW(1) .EQ.0

a complete duplication of the input, along with the results. This tape is used as input for the program MODPLOT.

On line printer (108)

the input parameters, and numerical tabulations and plots of the results. After each polygon the contribution of that polygon is printed and plotted, if desired.

The values tabulated are:

K - the field point number  
 FX(K), FZ(K) - the coordinates of the field point  
 ANOMALY - the anomaly contribution in mgals., and a blank column containing the weight contribution in kg.

After all polygons have been computed we get the RMS difference between the observed and calculated gravity, ignoring those points for which observed gravity is not known.

Then comes the number of points which are used in the computation of the RMS value.

If the boundary is to be varied these values appear for each iteration.

After the final new polygon has been determined the program prints out the new polygon points.

Then comes the tabulation of the final results

K, FX(K), FZ(K), as above

ANOMALY - unreferenced gravity anomaly  
 CALC REF- the calculated value referenced to the appropriate field point  
 RESIDUAL- the difference between the calculated reference value and the observed value  
 OBS ANOMALY - observed gravity  
 WEIGHT - the weight of a cm<sup>2</sup> column down to the bottom of the model  
 WGT DIFF - the weight -RWGT  
 WEIGHTTEST - this column will be a constant number provided the bottom of the model is flat and there are no holes or overlaps of the polygon. If this column is not constant then one of the above conditions has occurred, and there is probably an error in one or more of the polygons.

OUTPUT (continued)

Then follows a line printer plot of the crustal weight, and a line printer plot of the observed, and theoretical gravity.

Onto punch (106) - new polygons points if these are calculated.

USAGE:

Input is assumed to be on cards - assign F:1 to card reader; F:2 to output device.

A crustal model is prepared which is composed of various polygons of various densities. The polygons are numbered arbitrarily, with the exception of polygon number 1, which is a water layer, and the final polygon; which must be number 99. One point of each polygon is designated the "starting point". Points define the polygons by proceeding clockwise from the starting point and ending exactly at the same point. Polygon points must be arranged in this order for input to the TALPLOT program.

RESTRICTIONS:

1. Input is assumed to be on cards
2. There can be no more than
  - 200 field points per model
  - 150 polygon points per polygon
  - 20 variable polygon points
3. Polygon points must be arranged in order clockwise from starting point, and the final point of each polygon must be exact by the same as the starting point. Holes or overlaps between polygons will cause strange and exciting gravity variations. To check for these, see column marked WEIGHTEST in the line printer output.

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: PLOTA, PLOTTER, ISW, SIMUL, WEIG2

TIMING: Unknown

ERROR AND DIAGNOSTICS: None

PROGRAMMER: Allin Folinsbee

ORIGINATOR: Allin Folinsbee

DATE: 20 July 1975

```

1.  C  PROGRAM ABSTGC
2.  C  PROGRAM TO ABSTRACT GCBN DATA BY 10 MIN AVEREAGES
3.  C  VERSION OF 5 NOV 75 TO CORRECT CENTER LAT AND LONG
4.  C  ORIGINAL VERSION 20 AUG 75
5.  DIMENSION IBUFIN(6600)
6.  DIMENSION IBUFOT(1184)
7.  DIMENSION KODE(300),KDLAT(300),KLATMIN(300)
8.  DIMENSION KLATDEC(300),KLONGDEC(300)
9.  DIMENSION KDLONG(300),KLONGMIN(300),KFA(300),KELDEP(300)
10. DIMENSION KAVFA(6,6),NPTS(6,6),KAVEL(6,6)
11. DIMENSION H0WFA(6,6),KCENLAT(6,6),KCENLONG(6,6)
12. DIMENSION KCENEL(6,6),KCENFA(6,6)
13. C  INITIALIZATION
14. 100 CONTINUE
15.  OUTPUT 'ABSTGC VERSION OF 5 NOV 75'
16.  INITSW=ISW(-2)
17.  ITAPE=1
18.  JTAPE=2
19.  NDLT=100
20.  NCARD=105
21.  NBR=C
22.  NPRINT=108
23.  KURLAT=999
24.  KURLONG=999
25.  DO 108 J=1,6
26.  DO 107 I=1,6
27.  KAVFA(I,J)=0
28.  KAVEL(I,J)=0
29.  NPTS(I,J)=C
30.  KCENEL(I,J)=0
31.  KCENFA(I,J)=0
32.  H0WFA(I,J)=999
33.  KCENLAT(I,J)=0
34.  KCENLONG(I,J)=0
35. 107 CONTINUE
36. 108 CONTINUE
37.  NIN=0
38.  NI=0
39.  IF(ISW(46).EQ.C) GO TO 200
40. C  READ BOUNDS
41.  READ(NCARD,1003) KDTOP,KDBOT,KDLFT,KDRGT
42.  KDTOP=KDTOP+200
43.  KDBOT=KDBOT+200
44.  KDLFT=KDLFT+200
45.  KDRGT=KDRGT+200
46.  BUTFLT KDTOP,KDBOT,KDLFT,KDRGT
47. 1003 FORMAT(20I5)
48. 110 CONTINUE
49.  IF(ISW(46).NE.2) GO TO 200
50. C  DLT I/P LOGIC
51.  IF(NBR.EQ.C) READ(NDLT,1004,END=900) NBR,NDLTLT,NDLTLGR,
52.  NDLTLGL
53.  BUTFLT NBR,NDLTLT,NDLTLGR,NDLTLGL
54. C  CHECK BOUNDS
55. 1004 FORMAT(5X,I6,3I3)
56.  IF((NDLTLT.GT.KDTOP).OR.(NDLTLT.LT.KDBOT)) GO TO 150
57.  IF((NDLTLGL.LT.KDLFT).AND.(NDLTLGR.LT.KDLFT)) GO TO 150
58.  IF((NDLTLGL.GT.KDRGT).AND.(NDLTLGR.GT.KDRGT)) GO TO 150
59.  GO TO 200

```

```

60. C OUTSIDE BOUNDS
61. 150 CONTINUE
62.   BUTPLT NBR
63.   IF(NBR.LT.(300-NIN)) NIN=NIN+NBR/NBR=0; GO TO 110
64.   NBR=NBR-(300-NIN)
65.   CALL BUFFER IN(ITAPE,0,IBUFIN(1),1650,IKEY,NI)
66. 151 CONTINUE
67.   GO TO (152,155,153,154) IKEY
68. 152 BUTPLT 'WAITING'; GO TO 151
69. 153 BUTPLT 'EOF WHILE SKIPPING'; GO TO 900
70. 154 BUTPLT 'ERROR WHILE SKIPPING' ; GO TO 900
71. 155 CONTINUE
72.   NIN=0
73.   GO TO 150
74. C INPUT LOGIC
75. 200 CONTINUE
76.   IF(NBR.EQ.C.AND.ISW(46).EQ.2) GO TO 110
77.   IF(NIN.LT.NI) GO TO 220
78.   BUTPLT NIN
79. 210 CONTINUE
80.   CALL BUFFER IN(ITAPE,0,IBUFIN(1),1650,IKEY,NI)
81.   GO TO (211,215,213,214) IKEY
82. 211 BUTPLT 'WAITING'; GO TO 210
83. 213 BUTPLT 'EOF ON INPLT'; GO TO 900
84. 214 BUTPLT 'ERROR ON INPUT' ; GO TO 900
85. 215 CONTINUE
86.   NI=(NI*4)/22
87.   DECODE(NI*22,1001,IBUFIN(1),ND)
88.   * (KODE(I),KCLAT(I),KCLATDEC(I),KDLONG(I),KLONGDEC(I),
89.   *   KELDEP(I),KFA(I),I=1,NI)
90. 1001 FORMAT (300(I1,I3,I2,I3,I2,I6,I5))
91.   NIN=0
92.   DO 218 I=1,300
93.     DLATMIN=(FLBAT(KCLATDEC(I))/100.)*60.
94.     KLATMIN(I)=IFIX(DLATMIN)
95.     DLONGMIN=(FLBAT(KLONGDEC(I))/100.)*60.
96.     KLONGMIN(I)=IFIX(DLONGMIN)
97. 218 CONTINUE
98. 220 CONTINUE
99.   NIN=NIN+1
100.   IF(ISW(46).NE.2) GO TO 225
101.   NBR=NBR+1
102.   IF((KCLAT(NIN).GT.KCLATP).OR.(KCLAT(NIN).LT.KCLATP)) GO TO 200
103.   IF((KDLONG(NIN).LT.KDLFTP).OR.(KDLONG(NIN).GT.KDLRGT)) GO TO 200
104. 225 CONTINUE
105.   IF(KURLAT.EQ.999) GO TO 265
106.   IF((KCLAT(NIN).EQ.KURLAT).AND.(KDLONG(NIN).EQ.KURLONG)) GO TO 300
107. C NEW DEGREE SQUARE
108. C
109. C PREPARING LAST DEGREE SQUARE FOR O/P
110. DO 260 I=1,6
111. DO 250 J=1,6
112. KAVFA(I,J)=KAVFA(I,J)/NPTS(I,J)
113. KAVEL(I,J)=KAVEL(I,J)/NPTS(I,J)
114. 250 CONTINUE
115. 260 CONTINUE
116. ENCODE(1184,1002,IBUFOT(1),NE)
117. * KURLAT,KURLONG,0,0,0,0,0,0,
118. * ((KCNEL(I,J),KCENFA(I,J),KCENLAT(I,J),KCENLONG(I,J),
119. *   KAVEL(I,J),KAVFA(I,J),NPTS(I,J),I=1,6),J=1,6)

```

```

120.      CALL BUFFER OUT(JTAPE,C,IEUFOT(1),296,JKEY,N0)
121.      265 CONTINUE
122.      KURLAT=KDLAT(NIN)
123.      KURLONG=KDLONG(NIN)
124.      1002 FORMAT(37(I6,I6,2I2,2I6,I4))
125.      C      CLEARING ARRAYS BEFORE NEW DEGREE SQUARE
126.      DO 280 I=1,6
127.      DO 270 J=1,6
128.      KAVFA(I,J)=0
129.      KAVEL(I,J)=0
130.      NPTS(I,J)=0
131.      KCENEL(I,J)=0
132.      KCENFA(I,J)=0
133.      H0WFAR(I,J)=999
134.      KCENLAT(I,J)=0
135.      KCENLONG(I,J)=0
136.      27C CONTINUE
137.      28C CONTINUE
138.      C      SAME DEGREE SQUARE
139.      30C CONTINUE
140.      I=FIX(FLBAT(KLATMIN(NIN))/10.0)+1
141.      J=FIX(FLBAT(KLONGMIN(NIN))/10.0)+1
142.      IF(KLATMIN(NIN).EQ.0) I=1
143.      IF(KLONGMIN(NIN).EQ.0) J=1
144.      KAVEL(I,J)=KAVEL(I,J)+KELDEP(NIN)
145.      KAVFA(I,J)=KAVFA(I,J)+KFA(NIN)
146.      NPTS(I,J)=NPTS(I,J)+1
147.      DIST=SQRT((KLATMIN(NIN)-(I*10-5))**2+(KLONGMIN(NIN)-(J*10-5))**2)
148.      IF(DIST.LT.H0WFAR(I,J)) H0WFAR(I,J)=DIST
149.      *      KCENLAT(I,J)=KLATDEC(NIN);
150.      *      KCENLONG(I,J)=KLONGDEC(NIN);
151.      *      KCENFA(I,J)=KFA(NIN);
152.      *      KCENEL(I,J)=KELDEP(NIN)
153.      GO TO 200
154.      90C CONTINUE
155.      STOP
156.      END

```



HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	631	C0277
CONSTANTS:	10	0000A
LOCAL VARIABLES:	10800	C2A30
TEMPS:	1	C0001
TOTAL PROGRAM:	11442	C2CB2

```

1.      OUTPUT ' ', 'PROGRAM CHART - VERSION OF 29 SEPTEMBER 1975'
2.      OUTPUT ' ', ' '
3.      C      PROGRAM CHART
4.      C *    PERMIT READING OF DDLS BLOCKED DATA TAPES USING SUB. GET02
5.      C      17 SEPTEMBER 1975
6.      C *    ALLOW FOR ANNOTATING EVERY N HOURS USING ISW(71)
7.      C *    ADD ABORTSET CALL AND JUMP TO READ BRACH (310S) 17 JUNE 1975
8.      C      VERSION OF 14 NOV 74, TO CORRECT MERIDIONAL PARTS
9.      C      FOR SPHERICAL PLANET
10.     C      VERSION OF 14 SEPT 1973, SSW 20 & 21 FOR SIZE CONTROL
11.     C      ON SEISMICITY ANOTATION IF SSW(16)=1
12.     C      VERSION OF 7 SEPT 1973, ADDING DATE CHECK TO SEISMICITY ANOT
13.     C      AND CHANGE SIZE OF SEISMICITY SYMBOLS THROUGH SSW 15
14.     C      PROGRAM PRODUCES MERCATOR CHARTS FROM FIXSE DATA, SEAG DATA, GSUM DATA,
15.     C      MBATR DATA, OR CALCM DATA, WITH VALUES PLOTTED BY POINTS OR ALONG TRACK.
16.     C      WITH TIME ANNOTATION OPTION, DATE IS WRITTEN AT EACH CHANGE OF DAY.
17.     C      THIS PROGRAM ALSO WILL PLOT AND ANNOTATE FROM TABULATIONS OF SEISMIC
18.     C      REFRACTION, EARTHQUAKE EPICENTER, VOLCANO, AND HEAT FLOW DATA.
19.     C
20.     C SUBROUTINES USED: GRID2, BLINE, WHR, STAT, ISW, SP0T2, CALCBMP ROUTINES, RETBY,
21.     C      VETBY, T0DAY, ARLIM, ENDIO, EVIL, SHTV, DMT0R, FIND, GETG,
22.     C      GETS, GETP, GETY, GETV, GETL, GETX, AN0V2, P0STAP, GET02
23.     C
24.     C1 PLOT LABEL
25.     C2 SENSE SWITCH OPTIONS
26.     C3 TIME INTERVAL = READS START AND END DATE. IF TAPE INPUT, TAPE CAN BE PRE-
27.     C      POSITIONED BY SPECIFYING RECORDS TO BE SKIPPED. IF NO END DATE SPECIFIED,
28.     C      NO TEST MADE FOR END DATE.
29.     C4 PLOT FORMATS 1. SCALE IN INCHES PER DEGREE LONGITUDE 2. CONNECT PLOTTED
30.     C      POINTS 3. PLOT EVERY NTH POINT 4. PLOTTING EVERY NTH GRID LINE
31.     C      5. PLOT NUMBER 6. VALUE TO BE ANNOTED 7. ANNOTE EVERY NTH POINT 8. FORMAT
32.     C5 PLOT FORMATS CONTINUED 1. =1 2. CHARACTER HEIGHT (*0.07 INCH) 3. INTEGER
33.     C      OR NON-INTEGER CHART BOUNDS 4. DIGITS AFTER DECIMAL POINT PLOTTED
34.     C6 PLOT BOUNDARIES - TOP, BOTTOM, LEFT, AND RIGHT EDGES - IN DEGREES AND MINUTES
35.     C
36.     C
37.     C SSW(0) UP TO NOT READ ANY INPUT DATA, PLOT ONLY GRID
38.     C SSW(1) UP TO DELETE DRAWING NDEG GRID LINES (IN GRID2)
39.     C SSW(2) UP IF NEXT PLOT WILL BE ON THE SAME GRID AS THIS PLOT.
40.     C      ALSO, UP TO ADD CURRENT VELOCITIES TO SHIP'S VELOCITY
41.     C      FOR JFMT = 2 AND BLOCK .GT. 0 (IN SUB. GET02)
42.     C SSW(3) UP TO ANNOTATE ONLY AT CHANGE OF DAY
43.     C SSW(4) =0 FOR NO MARK AT DATA POINT
44.     C      =1 FOR PLOTTING A CIRCLE AROUND DATA POINT
45.     C      =2 FOR PLOTTING A DOT AT DATA POINT
46.     C SSW(5) =0 TO MAKE DEGREE ANNOTATIONS INSIDE GRID (CHARACTER SIZE 0.07 INCH)
47.     C      =1 TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.21 INCH)
48.     C      =2 TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.35 INCH)
49.     C SSW(6) FOR MULTIPLOT RUNS, UP WILL PUT AN EGF BETWEEN PLOTS. USEFUL TO PDP-5
50.     C      OPERATOR IN THE EVENT OF MECHANICAL MALFUNCTION OF PEN.
51.     C SSW(7) UP TO ANNOTATE ON LEFT SIDE OF TRACK
52.     C      DOWN TO ANNOTATE ON RIGHT SIDE OF TRACK
53.     C SSW(8) UP TO SUPPRESS PLOTTING OF GRID
54.     C SSW(9) UP TO LIST DATE AND TIME OF DATA OUT OF CHART BOUNDARIES
55.     C SSW(10) UP TO CALL SUBROUTINE MOUNT WHICH READS IN SERIAL NUMBER OF INPUT TAPE
56.     C      USED FOR PLOTS HAVING TWO OR MORE INPUT TAPES
57.     C SSW(11) UP TO ANNOTATE POINTS ALTERNATELY ON LEFT AND RIGHT SIDES OF TRACK
58.     C SSW(12) UP TO LIST DATE OF DATA JUST READ FOR IDENTIFICATION
59.     C SSW(13) UP IF TWO OR MORE PLOTS ARE BEING MADE FROM THE SAME TAPE AND THIS IS

```

```

60. C      NOT THE LAST PLOT. TAPE WILL BE POSITIONED AT BEGINNING OF CURRENT
61. C      FILE EVEN IF END-OF-FILE PASSED.
62. C      SSW(15) = N FOR ADDITIONAL SIZE INCREMENT IN PLOTTING SYMBOLS IF
63. C          SSW(16)=1
64. C      SSW(16) UP TO PLOT SPOT FOR SEISMICITY DATA PROPORTIONAL TO MAG AND DEPTH
65. C      SSW(17) = 0 TO PLOT AN X FOR SEISMICITY DATA PRE-1961
66. C          = 1 FOR PRE-1961 SEISMICITY DATA TO VARY ACCORDING TO MAG AND DEPTH
67. C          = 9 NOT TO PLOT PRE-1961 DATA
68. C      SSW(18) = 0 TO MAKE ANNOTATION AT RIGHT ANGLES TO INCREMENTAL TRACK (ANOV2)
69. C          = 1 TO MAKE ANNOTATIONS HORIZONTALLY
70. C          = 2 TO MAKE ANNOTATIONS VERTICALLY
71. C          = 3 TO INVERT ANNOTATIONS FOR HEADINGS 180 TO 269
72. C      SSW(19) = 0 FOR EARTH MERIDIONAL PARTS FROM BOWDITCH
73. C          = 1 FOR MERIDIONAL PARTS FOR SPHERICAL PLANET
74. C      SSW(20) = N, (SEISMICITY) FOR ADDITIONAL SIZE INCREMENT FOR ALL DATA PTS.
75. C      SSW(21) = N, (SEISMICITY) FOR SIZE FACTOR BY WHICH PLOTTING SYMBOLS WILL
76. C          VARY ACCORDING TO MAGNITUDE. IF N=0, THEN ANOV4 SETS N=2.
77. C      SSW(25) = 1 TO READ DATA IN ASCII CODING RATHER THAN EBCDIC
78. C          (FOR JFMT = 1, 4, AND 5 WITH BLOCK = 0)
79. C      SSW(71) = N TO ANNOTATE EVERY N HOURS ON THE HOUR
80. C
81. C      DIMENSION IBUF(1000), LABELP(20), N8W(4), IEND(4)
82. C      INTEGER BLOCK
83. C
84. C      IIN = 105
85. C      IIBUT = 108
86. C      IFMT=0
87. C      INYR=0
88. C      MFILE=0
89. C      CALL ABORTSET(310S,15)
90. C      PRINT DATE AND TIME OF JOB ON HEADING
91. C      CALL TODAY(N8W)
92. C      WRITE(IIBUT,13) N8W
93. C      13 FORMAT(1X,4A4)
94. C      WRITE(IIBUT,9)
95. C      9 FORMAT('PROGRAM CHART',9X,'ANNOTATED MERCATOR CHARTS USING FIXSE,
96. C          1SEAG, GSUM, MBATR, AND CALCM DATA FORMATS, //, OTHER OPTIONS PLOT S
97. C          2EIS REFR, SEISMICITY, VOLCANOES, HEAT FLOW, LUNAR, AND USER SUPPLI
98. C          3ED FORMAT')
99. C      CALL SETSKP(IND)
100. C      1C CALL PLOTS (IBUF, -1000)
101. C      MOVE PEN IN FROM EDGE AND ALONG PAPER SO THAT IF LETTERING OUTSIDE GRID WILL
102. C      NOT RUN OFF EDGE.
103. C      CALL PLOT(1.0,0.5,-3)
104. C      CALL STAT
105. C      79 IFLAG=0
106. C      CARD 1 PLOT LABEL (20A4)
107. C      PLOT LABEL ON LEFT MARGIN OF PLOT = SHIP, CRUISE, DATE, AND AREA. 80 CHAR.
108. C      READ(IIN,6,END=91,ERR=91) LABELP(I), I=1,20
109. C      6 FORMAT(20A4)
110. C      WRITE (IIBUT,7) LABELP(I), I=1,20
111. C      7 FORMAT(// 'PLOT LABEL: ',20A4)
112. C      CARD 2 SENSE SWITCH OPTIONS = SW1 TO SW79 IN COLUMNS 1 TO 79, SW0 IN COLUMN 80
113. C      INZ = ISW(-2)
114. C      IF (ISW(19).EQ.0) OUTPUT 'MERIDIONAL PARTS FOR EARTH FROM BOWDITCH'
115. C      IF (ISW(19).EQ.1) OUTPUT 'MERIDIONAL PARTS FOR A SPHERICAL PLANET'
116. C      ITAPE=1
117. C      IF ((ISW(10).EQ.1)) READ(IIN,6) NAME; CALL MBUNT(ITAPE,NAME);
118. C      1 WRITE(IIBUT,5784) NAME
119. C      5784 FORMAT('USING INPUT TAPE NUMBER' 1X,A4)

```

```

120. C POSITION LABEL DEPENDING WHETHER ANNOTATION INSIDE OR OUTSIDE GRID
121. PLACE=-0.80
122. IF (ISW(5).EQ.1) PLACE=-1.35
123. CALL SYMBOL(PLACE,1.0,0.14,LABELP,90.0,80)
124. C ANNOTATE DATE CHART MADE IN LOWER LEFT CORNER OF PLOT
125. CALL SYMBOL(PLACE,-0.48,0.07,NBW,0.0,16)
126. CALL PLOT(0.0,0.0,3)
127. C
128. CARD 3 TIME INTERVAL OF DATA, AND IF TAPE INPUT - POSITION ON TAPE.
129. 5 INIT=1
130. JSDA=0
131. JSMO=0
132. JSYR=0
133. JSHM=0
134. NY=1
135. NZ=1
136. NW=1
137. IEOD=0
138. IGAP=0
139. IAGAP=0
140. NPTS=0
141. IPCT=0
142. RADEG=57.29578
143. DEGRA=1.745329E-2
144. READ(IIN,2) ISTDA,ISTMO,ISTYR,ISTHM,IENDA,IENMO,IENYR,IENHM,ISKP,
145. 1 ISFIL,IBCKUP
146. 2 FORMAT(3I2,I4,5X,3I2,I4,5X,3I5)
147. WRITE(IIOU,6365)ISTDA,ISTMO,ISTYR,ISTHM,IENDA,IENMO,IENYR,
148. 1 IENHM,ISKP,ISFIL
149. 6365 FORMAT('START DATE ',3I2,1X,I4,5X,'END DATE ',3I2,1X,I4,5X,'RECORD
150. 1S SKIPPED TO START OF INTERVAL (ISKP): ',I4,2X,'FILES SKIPPED TO
151. 2START OF INTERVAL (ISFIL): ',I4)
152. IF(IBCKUP.NE.0) WRITE(IIOU,17) IBCKUP
153. 17 FORMAT(' SERIES OF OVERLAPPING CHARTS BEING MADE, OVERLAP OF NEXT
154. 1 CHART ON THIS CHART WILL BE ',I4,' RECORDS.')
```

C SPACING FILES ON MULTIFILE INPUT TAPES

```

156. IF(ISFIL.EQ.0) GO TO 11
157. CALL SKPFIL(ITAPE,ISFIL,IFWD)
158. GO TO (997,11,11,997,997,997) IND
159. C SPACING RECORDS ON INPUT TAPE
160. 11 IF(ISKP.EQ.0) GO TO 8
161. IF(ISKP.GT.0) CALL SKPREC(ITAPE,ISKP,IFWD)
162. IF(ISKP.LT.0) ISKP=-ISKP; CALL SKPREC(ITAPE,ISKP,'REV')
163. GO TO (999,8,999,999,999,999) IND
164. CARD 4 PLOT FORMAT PARAMETERS
165. 8 READ(IIN,3)SINCH,ITRK,LCNT,NDEG,NUMPL,NPTA,JFMT,NX,NFILE
166. 3 FORMAT(F10.0,3I5,1X,A4,4I5)
167. IF(JFMT.EQ.1) IFMT=4
168. IF(JFMT.EQ.2) IFMT=1
169. IF(JFMT.EQ.4) IFMT=3
170. IF(JFMT.EQ.5) IFMT=2
171. IF(IFMT.EQ.0) OUTPUT: 1,1 READ ROUTINE IS NOT GET02',1,1
172. WRITE(IIOU,6498) SINCH,ITRK,LCNT,NDEG,NUMPL,NPTA,JFMT,NX
173. 6498 FORMAT('CHART SCALE (SINCH) =',F7.3,' INCHES PER DEGREE LONGITUDE',
174. 1,4X,'TRACK POINTS CONNECTED (ITRK):',I2,4X,'PLOTING EVERY NTH POI
175. 2NT (LCNT):',I2,2X,'PLOTING EVERY NTH DEGREE LINE (NDEG):',I2,4X,
176. 3'PLOT NUMBER (NUMPL):',I4,4X,'ANNOTATE EVERY NTH PLOTTED POINT (NP
177. 4TA):',I2,2X,'DATA FORMAT (JFMT):',I2,4X,'VALUE ANNOTATED (NX):',
178. 5I2)
179. WRITE(IIOU,6493) NFILE

```

```

180. 6493 FORMAT(1 NUMBER OF FILES BEING OUTPUTTED ON SAME GRID (NFILE):'I3)
181. C      JFMT SPECIFIES THE FORMAT OF THE DATA
182. C      1 ... FIX DATA
183. C      2 ... SEAG1 DATA
184. C      JFMT 3 ... GSUM DATA
185. C      4 ... MBATR DATA
186. C      5 ... CALCM DATA
187. C      6 ... STATN DATA
188. C      7 ... SPFMT DATA
189. C      9 ... VOLCANOES
190. C     10 ... HEAT FLOW
191. C     11 ... LUNAR DATA
192. C     12 ... FLEXIBLE (SUPPLIED BY USER)
193. C
194. C      NSKIP=LCNT-1
195. C      NSTOP=LCNT
196. C      MSTOP=NPTA
197. CARD 5  PLOT FORMAT PARAMETERS
198. C      READ(IIN,4) KPT,KHT,ICTYP,IDEC,BLOCK
199. C      4 FORMAT(5G)
200. C      WRITE(IIOUT,6499) KPT,KHT,ICTYP,IDEC, BLOCK
201. 6499 FORMAT('MAGNIFICATION FACTOR (KPT)='I,12,4X,'CHARACTER HEIGHT *0.07
202. 1 INCH (KHT)='I,12,4X,'NON-INTEGER OR INTEGER CHART BOUNDS (ICTYP):'
203. 2,I2,2X,'DECIMAL POINT IN ANNOTATION (IDEC):'I,12,
204. 3 ' USING THE DDLS BLOCKED TAPES (NB/YES)='I,10,1):'I,F3,0/)
205. C
206. C      IDEC=N FOR N DIGITS TO RIGHT OF DECIMAL PT. IN SUBR ANOV2 ANNOTATION
207. C          =0  DECIMAL PT ONLY
208. C          -1  SUPPRESS DECIMAL POINT
209. C
210. C      ZZ = KPT
211. C      SINCH = SINCH * ZZ
212. C      ZHT=KPT*KHT
213. CARD 6  PLOT BOUNDARIES          (CARDS 6 TO 9 IF NON-INTEGER BOUNDS)
214. C      ITOP, IBOT, ILEFT, IRIGHT READ IN BY RETBY
215. C      WRITE(IIOUT,6364)
216. 6364 FORMAT('CHART BOUNDARIES:'I,20X,'EAST AND NORTH POSITIVE')
217. C      IF (ICTYP) 205,205,210
218. 210  CALL RETBY
219. C      A( IDATA,IEBD,IIN,IIOUT,ITAPE,NUMPL,DATA,RLAT,RLONG,KOGHM,IAGAP,LCN
220. BT,RADEG,DEGRA,KDEG2,IDEG2,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,RLEFT,
221. C ILEFT,RRIGT,IRIGT,SLTK,SLGK,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,NDEG,
222. D SLAT,SLONG,BOTMP,XX,YY,INIT,XOLD,YOLD)
223. C      GO TO 230
224. C      DEGREES AND MINUTES FOR RTOP, RBOT, RLEFT, RRIGHT READ IN BY
225. C      ARLIM AS CALLED BY VETBY
226. 205  CALL VETBY
227. C      A( IDATA,IEBD,IIN,IIOUT,ITAPE,NUMPL,DATA,RLAT,RLONG,KOGHM,IAGAP,LCN
228. BT,RADEG,DEGRA,KDEG2,IDEG2,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,RLEFT,
229. C ILEFT,RRIGT,IRIGT,SLTK,SLGK,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,NDEG,
230. D SLAT,SLONG,BOTMP,XX,YY,INIT,XOLD,YOLD)
231. 230  CONTINUE
232. C
233. C *  INITIALIZE GET02
234. C
235. C      IF(JFMT.EG.1).OR.
236. $      JFMT.EG.4).OR.
237. $      JFMT.EG.5).OR.
238. $      BLOCK.GT.0
239. $      ) CALL GET02(ITAPE,0,
    JDA,M0,JYR,JHM,RLAT,RLONG,NX,NY,NZ,NW,

```

```

240.      * DATA, DATAY, DATAZ, DATAW, ITRK, LCNT, NPTA, IDEC,
241.      * IE0D, IAGAP, IFMT, INYR, BLOCK)
242.      IF (ISW(19).EQ.1) GO TO 240
243.  C      T9 SET ORIGIN - MERIDIONAL PARTS FROM BOWDITCH
244.      A = ABS(RB0T)/2.0*(45.0*DEGRA)
245.      B = (ALOG(SIN(A)/COS(A))) * 0.4342945
246.      B0TMP = 7.915704E+03 * B - (23.26893*SIN(ABS(RB0T
247.      1))) - (0.0525*(SIN(ABS(RB0T)))*3)
248.      GO TO 15
249.  C 240 CONTINUE
250.  C      T9 SET ORIGIN - MERIDIONAL PARTS FOR SPHERICAL PLANET
251.      A = ABS(RB0T)/2.0*(45.0*DEGRA)
252.      B = (ALOG(SIN(A)/COS(A)))
253.      B0TMP = 3437.747*B
254.      15 IF (RB0T) 20,30,30
255.      20 B0TMP = -B0TMP
256.      30 IF (ICTYP) 22,22,32
257.  C      SUBROUTINE 0LINE FOR GRID ENCLOSED BY NON-INTEGER DEGREES
258.      22 CALL 0LINE(ZZ,ZHT,
259.      A IDATA, IE0D, IIN, IIBUT, ITAPE, NUMPL, DATA, RLAT, RLONG, K0GHM, IAGAP, LCN
260.      BT, RADEG, DEGRA, KDEG2, IDEG2, FDEG2, RDEG2, RT0P, IT0P, RB0T, IB0T, RLEFT,
261.      C ILEFT, RRIGHT, IRIGHT, SLTK, SLGK, SINCH, SMP, FB0T, FT0P, FLEFT, FRIGT, NDEG,
262.      D SLAT, SLONG, B0TMP, XX, YY, INIT, X0LD, Y0LD)
263.  C      CHECKING IF ONLY DRAWING OF GRID WANTED
264.      IF (ISW(0)) 310,310,91
265.  C
266.  C      SUBROUTINE GRID2 FOR GRID ENCLOSED BY WHOLE DEGREES
267.      32 CALL GRID2(ZZ,ZHT, NUMPL, DEGRA, FDEG2, RDEG2, RT0P, IT0P, RB0T, IB0T,
268.      1 RLEFT, ILEFT, RRIGHT, IRIGHT, SINCH, SMP, FB0T, FT0P, FLEFT, FRIGT, NDEG,
269.      2 SLAT, SLONG, B0TMP)
270.  C      CHECKING IF ONLY DRAWING OF GRID WANTED
271.      IF (ISW(0)) 310,310,91
272.  C
273.  C END OF INITIALIZATION, BEGIN PLOTTING POINTS
274.  C
275.  C CHECKING IF TRACK POINTS SHOULD BE CONNECTED
276.      35 IF (ITRK) 40,40,45
277.      40 IP = 3
278.      GO TO 70
279.      45 IF (INIT) 55,55,50
280.      50 IP = 3
281.      GO TO 70
282.      55 IF (IAGAP) 65,65,60
283.      60 IP = 3
284.      GO TO 70
285.      65 IP = 2
286.  C PLOT DATA POINT
287.      70 SLAT = RLAT
288.      SLONG = RLONG
289.      CALL WHR(DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, B0TMP, XX, YY)
290.      CALL PLOT(XX, YY, IP)
291.      IPCT = IPCT + 1
292.      IF (ISW(16)) 73,73,710
293.      710 CALL AN0V4(XX, YY, DATAZ, DATAW, JYR)
294.      GO TO 72
295.  C CHECK FOR TYPE OF DATA POINT ANNOTATION MADE IN SP0T2
296.      73 CALL SP0T2(XX, YY)
297.      74 DATA = DATAX
298.      IF (NX) 71,72,71
299.      71 MST0P = MST0P - 1

```

```

300.      IF(MSTOP)871,871,72
301.      871 CONTINUE
302.      C
303.      C      TEST FOR TIME ANNOTATED AT 2 HOUR INTERVALS
304.      C
305.      IF(ISW(71)) 872,878,872
306.      872 IF(JDA = J0DA) 875,873,875
307.      873 IF(FLBAT(JHM/100) = FLBAT(J0HM/100)) 874,72,874
308.      874 ITCT=ITCT + 1
309.      IF(ITCT .LT. ISW(71)) GO TO 72
310.      875 ITCT=0
311.      878 CONTINUE
312.      CALL ANGV2(ZZ,ZHT,NX,JDA,JMB,JYR,JHM,J0DA,XX,YY,INIT,DATA,IDEC)
313.      MST0P=NPTA
314.      72 INIT = 0
315.      RLTO = RLAT
316.      RLGO = RLONG
317.      J0DA=JDA
318.      J0MB=JMB
319.      J0YR=JYR
320.      J0HM=JHM
321.      NPTS=NPTS+1
322.      C DATA POINT PLOTTED, INPUT NEXT DATA POINT
323.      C INPUT DATA FORMATS: 308 FIXSE 308 SEAG1 311 GSUM 308 MBATR 308 CALCM
324.      C 314 STATN 340 SPFMT 342 SEISMICITY 344 VOLCANOES 346 HEAT FLOW
325.      C 348 LUNAR DATA 350 FLEXIBLE (USER SUPPLIED)
326.      31C GO TO (308,308,311,308,308,314,340,342,344,346,348,350) JFMT
327.      308 CONTINUE
328.      IF(BLOCK .EQ. 0 .AND. JFMT .EQ. 2) CALL GETS(ITAPE,
329.      $ NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,RLAT,RLONG,
330.      $ JDA,JMB,JYR,JHM,IE0D) ; GO TO 78
331.      CALL GETO2(ITAPE,1,
332.      $ JDA,JMB,JYR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
333.      $ DATAZ,DATAY,DATAZ,DATAW,ITRK,LCNT,NPTA,IDEC,
334.      $ IE0D,IAGAP,IFMT,INYR,BLOCK)
335.      GO TO 78
336.      311 CALL GETG (ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,
337.      1 RLAT,RLONG,JDA,JMB,JYR,JHM,IE0D)
338.      GO TO 78
339.      314 CALL GETST(ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,
340.      1 RLAT,RLONG,JDA,JMB,JYR,JHM,IE0D)
341.      GO TO 78
342.      34C CALL GETP (ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,RLAT,RLONG,
343.      1 JDA,JMB,JYR,JHM,IE0D)
344.      GO TO 78
345.      342 CALL GETY (ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,RLAT,RLONG,
346.      1 JDA,JMB,JYR,JHM,IE0D)
347.      GO TO 78
348.      344 CALL GETV (ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,
349.      1 RLAT,RLONG,JDA,JMB,JYR,JHM,IE0D)
350.      GO TO 78
351.      346 CALL GETH (ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,
352.      1 RLAT,RLONG,JDA,JMB,JYR,JHM,IE0D)
353.      GO TO 78
354.      348 CALL GETL (ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,
355.      1 RLAT,RLONG,JDA,JMB,JYR,JHM,IE0D)
356.      GO TO 78
357.      35C CALL GETX (ITAPE,NX,NY,NZ,NW,DATAZ,DATAY,DATAZ,DATAW,
358.      1 RLAT,RLONG,JDA,JMB,JYR,JHM,IE0D)
359.      C

```

```

360.      78 IF (IE0D-1) 85,1212,1212
361.      1212 NFILE=NFILE+1
362.          IF(NFILE.LE.0) GO TO 12
363.          INIT = 1
364.          IE0D=0
365.          MFILE=MFILE+1
366.          OUTPUT 'PLOTING CONTINUES FROM NEXT FILE'
367.          GO TO 310
368.      C
369.      C END OF DATA FILE - CHECK SENSE SWITCHES FOR NEXT STEP
370.      C
371.      C CHECKING IF ANOTHER PLOT FOLLOWS OR TO EXIT
372.      12 IF (ISW(2).EQ.0.AND.ISW(13).EQ.0.AND.IBACK.EQ.0) GO TO 91
373.          IF (ISW(2).EQ.1) CALL PLOT(0.0,0.0,3)
374.      C SENSE SWITCH 13 CHECKS WHETHER ANOTHER PLOT IS TO BE MADE FROM SAME FILE
375.          IF (ISW(13)) 80,81,80
376.      80 IBACK=0
377.          IF (IE0D.EQ.1) IBACK=1
378.          IF (MFILE.NE.0) IBACK=MFILE+1
379.          CALL SKPFIL(ITAPE,IBACK,IREV)
380.      C SEQUENTIAL PLOTS FROM SAME FILE WITH (IBCKUP) OVERLAP
381.      81 IF (IBCKUP.EQ.0) GO TO 96
382.          IBCKUP=IBCKUP+1
383.          IF (IE0D.EQ.0) CALL SKPREC(ITAPE,IBCKUP,IREV)
384.          IF (IE0D.EQ.1) CALL SKPFIL(ITAPE,1,IREV)
385.      C ESTABLISH ORIGIN OF NEW PLOT
386.      96 IF (ISW(2).EQ.1) GO TO 95
387.          XINCRA=ABS(RDEG2)*RADEG*SINCH+4.00
388.          CALL PLOT(XINCRA,0.0,-3)
389.      95 IF (ISW(6).EQ.1) CALL PLOT(-1.00,-0.50,999); GO TO 94
390.      C PUT RUN AND DATA CARDS FOLLOWING LAST DATA DECK TO REINITIALIZE PROGRAM.
391.          GO TO 79
392.      C CLOSE PLOT TAPE AND END-OF-JOB LABEL FOR PDP-5 OPERATOR
393.          DATA (IEND(I),I=1,4) /'END OF CHART JOB'/
394.      91 XINCRA=ABS(RDEG2)*RADEG*SINCH+1.00
395.      C COMPLETE BOTTOM AND RIGHT SIDES OF FIDUCIAL HALF-INCH SQUARE DRAWN IN GRID2
396.          CALL PLOT(XINCRA,-0.5,-3)
397.          CALL PLOT(0.5,0.0,-2)
398.          CALL PLOT(0.0,0.5,-2)
399.          CALL PLOT(2.5,-0.5,-3)
400.          CALL SYMBOL(0.0,0.0,0.42,IEND,90.0,16)
401.          CALL PLOT(4.0,0.0,999)
402.      94 IF (ISW(10).EQ.1) CALL PREL(1)
403.          WRITE(IIOUT,97)IPCT
404.      97 FORMAT(' NUMBER POINTS PLOTTED = ',I8)
405.          CALL EXIT
406.      C
407.      C SKIPPING POINTS, IF EVERY POINT NOT TO BE PLOTTED.
408.      85 IF (NSKIP) 186,186,185
409.      185 CALL SKPREC(ITAPE,NSKIP)
410.          GO TO (999,186,1212,999,1212,999) IND
411.      C CHECK IF DATE IS WITHIN SPECIFIED TIME INTERVAL
412.      C IFLAG IS A FLAG TO ALLOW SKIPPING THE FIRST CALL TO FIND
413.      C IF WE HAVE ALREADY FOUND THE STARTING DATE
414.      186 IF (IFLAG.NE.0) GO TO 82
415.          CALL FIND(ISTDA,ISTMO,ISTYR,ISTHM,JDA,JMO,JYR,JHM,INDK)
416.          IF (INDK.EQ.-1) GO TO 310
417.          IFLAG=1
418.      82 IF (IENYR.EQ.0) GO TO 851
419.          CALL FIND(IENDA,IENMO,IENYR,IENHM,JDA,JMO,JYR,JHM,INDK)

```

```

420.      IF(INDK.EQ.1) GO TO 995
421.  C   CHECKING IF DATA WITHIN CHART BOUNDRIES
422.      851  NSTOP=LCNT
423.          IF(RTOP-RLAT) 100,100,86
424.      86  IF(RLAT-RBOT) 100,88,98
425.      88  IF(RLONG-RLEFT) 100,90,90
426.      90  IF(RRIGHT-RLONG) 100,100,35
427.  C   DATA OUTSIDE OF BOUNDS
428.  C   WRITING OUT DATE IF DATA POINT IS OUT OF BOUNDS
429.      100  INIT=1
430.          IF(ISH(9))410,310,410
431.      410  WRITE(IIOUT,420)JDA,JM0,JYR,JHM
432.      420  FORMAT('00B ',3I2,1X,I4)
433.          GO TO 310
434.      995  WRITE(IIOUT,996) JDA,JM0,JYR,JHM
435.      996  FORMAT('END DATE PASSED',2X,3I2,1X,I4)
436.          GO TO 12
437.  C   ERROR MESSAGES IF MISTAKE IN TAPE FILE OR RECORD SPACING.
438.      999  WRITE(IIOUT,998) IND
439.      998  FORMAT('ERROR IN SKPREC, IND=',I2)
440.          GO TO 91
441.      997  WRITE(IIOUT,994) IND
442.      994  FORMAT('ERROR IN SKPFIL, IND=',I2)
443.          GO TO 91
444.      END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
A	R	SCALR	00464	1	ABORTSET	SPRGG	SPRGG	00464	1	ABS	R	SPRGG	00457	1	ABN4	R	SPRGG	00457	1
ALOG	R	SPRGG	00465	1	ABV2	SPRGG	SPRGG	00465	1	EXTRN	EXTRN	EXTRN	00458	1	B0THP	R	SCALR	00458	1
B	R	SPRGG	00466	1	BLACK	I	SCALR	00466	1	EXTRN	EXTRN	EXTRN	00459	1	DATAW	R	SCALR	00459	1
COS	R	SPRGG	00467	1	DATA	R	SCALR	00467	1	EXTRN	EXTRN	EXTRN	00460	1	DATZ	R	SCALR	00460	1
DATA X	R	SCALR	00468	1	EXIT	R	SPRGG	00468	1	EXTRN	EXTRN	EXTRN	00461	1	F0T	R	SCALR	00461	1
DEGRA	R	SCALR	00469	1	FIND	R	SPRGG	00469	1	EXTRN	EXTRN	EXTRN	00462	1	F0T	R	SCALR	00462	1
FDEG2	R	SCALR	00470	1	FRIGT	R	SPRGG	00470	1	EXTRN	EXTRN	EXTRN	00463	1	F0T	R	SCALR	00463	1
FLOAT	R	SCALR	00471	1	GET1	R	SPRGG	00471	1	EXTRN	EXTRN	EXTRN	00464	1	F0T	R	SCALR	00464	1
GETP	R	SPRGG	00472	1	GETS	R	SPRGG	00472	1	EXTRN	EXTRN	EXTRN	00465	1	F0T	R	SCALR	00465	1
GETV	R	SPRGG	00473	1	GETX	R	SPRGG	00473	1	EXTRN	EXTRN	EXTRN	00466	1	F0T	R	SCALR	00466	1
GETO2	R	SPRGG	00474	1	GRID2	R	SPRGG	00474	1	EXTRN	EXTRN	EXTRN	00467	1	F0T	R	SCALR	00467	1
IAGAP	R	SCALR	00475	1	IBACK	R	SPRGG	00475	1	EXTRN	EXTRN	EXTRN	00468	1	F0T	R	SCALR	00468	1
IBOT	R	SCALR	00476	1	IBUF	R	SPRGG	00476	1	EXTRN	EXTRN	EXTRN	00469	1	F0T	R	SCALR	00469	1
IDATA	R	SCALR	00477	1	IDEC	R	SPRGG	00477	1	EXTRN	EXTRN	EXTRN	00470	1	F0T	R	SCALR	00470	1
IEND	R	SCALR	00478	1	IDEN2	R	SPRGG	00478	1	EXTRN	EXTRN	EXTRN	00471	1	F0T	R	SCALR	00471	1
IENH	R	SCALR	00479	1	IENYR	R	SPRGG	00479	1	EXTRN	EXTRN	EXTRN	00472	1	F0T	R	SCALR	00472	1
IFLAG	R	SCALR	00480	1	IFPT	R	SPRGG	00480	1	EXTRN	EXTRN	EXTRN	00473	1	F0T	R	SCALR	00473	1
IIN	R	SCALR	00481	1	IFOUT	R	SPRGG	00481	1	EXTRN	EXTRN	EXTRN	00474	1	F0T	R	SCALR	00474	1
IND	R	SCALR	00482	1	INDK	R	SPRGG	00482	1	EXTRN	EXTRN	EXTRN	00475	1	F0T	R	SCALR	00475	1
INR	R	SCALR	00483	1	INZ	R	SPRGG	00483	1	EXTRN	EXTRN	EXTRN	00476	1	F0T	R	SCALR	00476	1
IPCT	R	SCALR	00484	1	IRGT	R	SPRGG	00484	1	EXTRN	EXTRN	EXTRN	00477	1	F0T	R	SCALR	00477	1
ISKP	R	SCALR	00485	1	ISTDA	R	SPRGG	00485	1	EXTRN	EXTRN	EXTRN	00478	1	F0T	R	SCALR	00478	1
ISTH	R	SCALR	00486	1	ISTYR	R	SPRGG	00486	1	EXTRN	EXTRN	EXTRN	00479	1	F0T	R	SCALR	00479	1
ITAPE	R	SCALR	00487	1	ITCT	R	SPRGG	00487	1	EXTRN	EXTRN	EXTRN	00480	1	F0T	R	SCALR	00480	1
ITRK	R	SCALR	00488	1	JDA	R	SPRGG	00488	1	EXTRN	EXTRN	EXTRN	00481	1	F0T	R	SCALR	00481	1
JHM	R	SCALR	00489	1	JMB	R	SPRGG	00489	1	EXTRN	EXTRN	EXTRN	00482	1	F0T	R	SCALR	00482	1
JYR	R	SCALR	00490	1	JMB2	R	SPRGG	00490	1	EXTRN	EXTRN	EXTRN	00483	1	F0T	R	SCALR	00483	1
K0GMH	R	SCALR	00491	1	KPT	R	SPRGG	00491	1	EXTRN	EXTRN	EXTRN	00484	1	F0T	R	SCALR	00484	1
K0GTH	R	SCALR	00492	1	KFILE	R	SPRGG	00492	1	EXTRN	EXTRN	EXTRN	00485	1	F0T	R	SCALR	00485	1
K0GTH	R	SCALR	00493	1	KFILE	R	SPRGG	00493	1	EXTRN	EXTRN	EXTRN	00486	1	F0T	R	SCALR	00486	1
K0GTH	R	SCALR	00494	1	KFILE	R	SPRGG	00494	1	EXTRN	EXTRN	EXTRN	00487	1	F0T	R	SCALR	00487	1
K0GTH	R	SCALR	00495	1	KFILE	R	SPRGG	00495	1	EXTRN	EXTRN	EXTRN	00488	1	F0T	R	SCALR	00488	1
K0GTH	R	SCALR	00496	1	KFILE	R	SPRGG	00496	1	EXTRN	EXTRN	EXTRN	00489	1	F0T	R	SCALR	00489	1
K0GTH	R	SCALR	00497	1	KFILE	R	SPRGG	00497	1	EXTRN	EXTRN	EXTRN	00490	1	F0T	R	SCALR	00490	1
K0GTH	R	SCALR	00498	1	KFILE	R	SPRGG	00498	1	EXTRN	EXTRN	EXTRN	00491	1	F0T	R	SCALR	00491	1
K0GTH	R	SCALR	00499	1	KFILE	R	SPRGG	00499	1	EXTRN	EXTRN	EXTRN	00492	1	F0T	R	SCALR	00492	1
K0GTH	R	SCALR	00500	1	KFILE	R	SPRGG	00500	1	EXTRN	EXTRN	EXTRN	00493	1	F0T	R	SCALR	00493	1
K0GTH	R	SCALR	00501	1	KFILE	R	SPRGG	00501	1	EXTRN	EXTRN	EXTRN	00494	1	F0T	R	SCALR	00494	1
K0GTH	R	SCALR	00502	1	KFILE	R	SPRGG	00502	1	EXTRN	EXTRN	EXTRN	00495	1	F0T	R	SCALR	00495	1
K0GTH	R	SCALR	00503	1	KFILE	R	SPRGG	00503	1	EXTRN	EXTRN	EXTRN	00496	1	F0T	R	SCALR	00496	1
K0GTH	R	SCALR	00504	1	KFILE	R	SPRGG	00504	1	EXTRN	EXTRN	EXTRN	00497	1	F0T	R	SCALR	00497	1
K0GTH	R	SCALR	00505	1	KFILE	R	SPRGG	00505	1	EXTRN	EXTRN	EXTRN	00498	1	F0T	R	SCALR	00498	1
K0GTH	R	SCALR	00506	1	KFILE	R	SPRGG	00506	1	EXTRN	EXTRN	EXTRN	00499	1	F0T	R	SCALR	00499	1
K0GTH	R	SCALR	00507	1	KFILE	R	SPRGG	00507	1	EXTRN	EXTRN	EXTRN	00500	1	F0T	R	SCALR	00500	1
K0GTH	R	SCALR	00508	1	KFILE	R	SPRGG	00508	1	EXTRN	EXTRN	EXTRN	00501	1	F0T	R	SCALR	00501	1
K0GTH	R	SCALR	00509	1	KFILE	R	SPRGG	00509	1	EXTRN	EXTRN	EXTRN	00502	1	F0T	R	SCALR	00502	1
K0GTH	R	SCALR	00510	1	KFILE	R	SPRGG	00510	1	EXTRN	EXTRN	EXTRN	00503	1	F0T	R	SCALR	00503	1
K0GTH	R	SCALR	00511	1	KFILE	R	SPRGG	00511	1	EXTRN	EXTRN	EXTRN	00504	1	F0T	R	SCALR	00504	1
K0GTH	R	SCALR	00512	1	KFILE	R	SPRGG	00512	1	EXTRN	EXTRN	EXTRN	00505	1	F0T	R	SCALR	00505	1
K0GTH	R	SCALR	00513	1	KFILE	R	SPRGG	00513	1	EXTRN	EXTRN	EXTRN	00506	1	F0T	R	SCALR	00506	1
K0GTH	R	SCALR	00514	1	KFILE	R	SPRGG	00514	1	EXTRN	EXTRN	EXTRN	00507	1	F0T	R	SCALR	00507	1
K0GTH	R	SCALR	00515	1	KFILE	R	SPRGG	00515	1	EXTRN	EXTRN	EXTRN	00508	1	F0T	R	SCALR	00508	1
K0GTH	R	SCALR	00516	1	KFILE	R	SPRGG	00516	1	EXTRN	EXTRN	EXTRN	00509	1	F0T	R	SCALR	00509	1
K0GTH	R	SCALR	00517	1	KFILE	R	SPRGG	00517	1	EXTRN	EXTRN	EXTRN	00510	1	F0T	R	SCALR	00510	1
K0GTH	R	SCALR	00518	1	KFILE	R	SPRGG	00518	1	EXTRN	EXTRN	EXTRN	00511	1	F0T	R	SCALR	00511	1
K0GTH	R	SCALR	00519	1	KFILE	R	SPRGG	00519	1	EXTRN	EXTRN	EXTRN	00512	1	F0T	R	SCALR	00512	1
K0GTH	R	SCALR	00520	1	KFILE	R	SPRGG	00520	1	EXTRN	EXTRN	EXTRN	00513	1	F0T	R	SCALR	00513	1
K0GTH	R	SCALR	00521	1	KFILE	R	SPRGG	00521	1	EXTRN	EXTRN	EXTRN	00514	1	F0T	R	SCALR	00514	1
K0GTH	R	SCALR	00522	1	KFILE	R	SPRGG	00522	1	EXTRN	EXTRN	EXTRN	00515	1	F0T	R	SCALR	00515	1
K0GTH	R	SCALR	00523	1	KFILE	R	SPRGG	00523	1	EXTRN	EXTRN	EXTRN	00516	1	F0T	R	SCALR	00516	1
K0GTH	R	SCALR	00524	1	KFILE	R	SPRGG	00524	1	EXTRN	EXTRN	EXTRN	00517	1	F0T	R	SCALR	00517	1
K0GTH	R	SCALR	00525	1	KFILE	R	SPRGG	00525	1	EXTRN	EXTRN	EXTRN	00518	1	F0T	R	SCALR	00518	1
K0GTH	R	SCALR	00526	1	KFILE	R	SPRGG	00526	1	EXTRN	EXTRN	EXTRN	00519	1	F0T	R	SCALR	00519	1
K0GTH	R	SCALR	00527	1	KFILE	R	SPRGG	00527	1	EXTRN	EXTRN	EXTRN	00520	1	F0T	R	SCALR	00520	1
K0GTH	R	SCALR	00528	1	KFILE	R	SPRGG	00528	1	EXTRN	EXTRN	EXTRN	00521	1	F0T	R	SCALR	00521	1
K0GTH	R	SCALR	00529	1	KFILE	R	SPRGG	00529	1	EXTRN	EXTRN	EXTRN	00522	1	F0T	R	SCALR	00522	1
K0GTH	R	SCALR	00530	1	KFILE	R	SPRGG	00530	1	EXTRN	EXTRN	EXTRN	00523	1	F0T	R	SCALR	00523	1
K0GTH	R	SCALR	00531	1	KFILE	R	SPRGG	00531	1	EXTRN	EXTRN	EXTRN	00524	1	F0T	R	SCALR	00524	1
K0GTH	R	SCALR	00532	1	KFILE	R	SPRGG	00532	1	EXTRN	EXTRN	EXTRN	00525	1	F0T	R	SCALR	00525	1
K0GTH	R	SCALR	00533	1	KFILE	R	SPRGG	00533	1	EXTRN	EXTRN	EXTRN	00526	1	F0T	R	SCALR	00526	1
K0GTH	R	SCALR	00534	1	KFILE	R	SPRGG	00534	1	EXTRN	EXTRN	EXTRN	00527	1	F0T	R	SCALR	00527	1
K0GTH	R	SCALR	00535	1	KFILE	R	SPRGG	00535	1	EXTRN	EXTRN	EXTRN	00528	1	F0T	R	SCALR	00528	1
K0GTH	R	SCALR	00536	1	KFILE	R	SPRGG	00536	1	EXTRN	EXTRN	EXTRN	00529	1	F0T	R	SCALR	00529	1
K0GTH	R	SCALR	00537	1	KFILE	R	SPRGG	00537	1	EXTRN	EXTRN	EXTRN	00530	1	F0T	R	SCALR	00530	1
K0GTH	R	SCALR	00538	1	KFILE	R	SPRGG	00538	1	EXTRN	EXTRN	EXTRN	00531	1	F0T	R	SCALR	00531	1
K0GTH	R	SCALR	00539	1	KFILE	R	SPRGG	00539	1	EXTRN	EXTRN	EXTRN	00532	1	F0T	R	SCALR	00532	1
K0GTH	R	SCALR	00540	1	KFILE	R	SPRGG	00540	1	EXTRN	EXTRN	EXTRN	00533	1	F0T	R	SCALR	00533	1
K0GTH	R	SCALR	00541	1	KFILE	R	SPRGG	00541	1	EXTRN	EXTRN	EXTRN	00534	1	F0T	R	SCALR	00534	1
K0GTH	R	SCALR	00542	1	KFILE	R	SPRGG	00542	1	EXTRN	EXTRN	EXTRN	00535	1	F0T	R	SCALR	00535	1
K0G																			

15	003A5	17	0017F	20	003A7	22	003AC	30	003AA	32	003E1
35	00401	40	00403	45	00406	50	00408	55	0040B	60	0040D
65	00410	70	00412	71	0043E	72	0046B	73	00436	74	0043A
78	00556	79	0007A	80	0058B	81	0059C	82	0062B	85	00608
86	00640	88	00643	90	00646	91	005C9	94	005EF	95	005BD
96	005AE	97	005FC	100	00649	105	0060A	186	00619	205	0031A
210	002EC	230	00347	240	00393	300	0048B	310	0047A	311	0048F
314	004D2	340	004E5	342	004F8	344	0050B	346	0051E	348	00531
350	00544	410	00650	420	00658	710	0042E	851	0063B	871	00441
872	00446	873	00449	874	00452	875	00458	878	0045A	994	00684
995	0065F	996	00667	997	0067F	998	00676	999	00671	1212	00559
5784	000E4	6364	002DC	6365	0014F	6493	00268	6498	00210	6499	00293

## LOCAL VARIABLES (1133 WORDS):

0000C	IBUF	003E8	LABELP	003FC	N8W	00400	IEND	00404	BL0CK	00405	IIN
00406	II0UT	00407	IFMT	00408	INR	00409	MFILE	0040A	IND	0040B	IFLAG
0040C	I	0040D	INZ	004CE	ITAPE	004CF	NAME	00410	PLACE	00411	INIT
00412	J8DA	00413	J8M8	00414	J8YR	00415	J8HM	00416	NY	00417	NZ
00418	NW	00419	TE8D	0041A	ICAP	0041B	IAGAP	0041C	NPTS	0041D	IPCT
0041E	RADEG	0041F	DEGRA	0042C	ISTDA	00421	ISTM8	00422	ISTYR	00423	ISTHM
0042A	IENDA	00425	IENM8	00426	IENYR	00427	IENHM	00428	ISKP	00429	ISFIL
0042A	IBCKUP	00428	SINCH	0042C	ITRK	0042D	LCNT	0042E	NDEG	0042F	NUMPL
0043C	NPTA	00431	VFMT	00432	NX	00433	NFILE	00434	NSKIP	00435	NST0P
00436	MST0P	00437	KPT	00438	KHT	00439	ICTYP	0043A	IDEC	0043B	ZZ
0043C	ZHT	0043D	IDATA	0043E	DATA	0043F	RLAT	00440	RL0NG	00441	K8GHM
00442	KDEG2	00443	IDEG2	00444	RDEG2	00445	RDEG2	00446	RT0P	00447	IT0P
00448	R88T	00449	I88T	0044A	RLEFT	0044B	ILEFT	0044C	RRIGT	0044D	IRIGT
0044E	SLTK	0044F	SLGK	0045C	SMP	00451	F88T	00452	FT0P	00453	FLEFT
00454	FRIGT	00455	SLAT	00456	SL0NG	00457	B8TMP	00458	XX	00459	YY
0045A	X8LD	0045B	Y8LD	0045C	JDA	0045D	JM8	0045E	JYR	0045F	JHM
00460	DATAX	00461	DATAY	00462	DATZ	00463	DATAM	00464	A	00465	B
00466	IP	00467	ITCT	00468	RLT8	00469	RLG8	0046A	IBACK	0046B	XINCRE

## BLANK COMMON (0 WORDS)

## INTRINSIC SUBPROGRAMS USED:

ABS	AL8G	C8S	FL8AT	SIN
-----	------	-----	-------	-----

## EXTERNAL SUBPROGRAMS REQUIRED:

ABORTSET	AN8V2	AN8V4	EXIT	FIND	GETL
GETP	GETS	GETST	GETV	GETX	GETH
ISL	M8UNT	MREL	GLINE	PL8T	GET02
SKPFFIL	SKPREC	SP8T2	STAT	SYMB8L	RETBY
F:101	F:1C2	F:103	F:104	F:105	VETBY
M:8C	9AL8G	9BCRDDEE	9BCDREAD	9BCDWRIT	F:108
918DATA	918LUSA	91T8R	9PRINT	9SIN	SENDI8L
					M:108
					9INITIAL

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	1678	0068E
CONSTANTS:	44	0002C
LOCAL VARIABLES:	1133	0046D
TEMPS:	3	00003
	-----	-----
TOTAL PROGRAM:	2858	00B2A

```

1.  C      PROGRAM CHARTG
2.  C
3.  C      VERSION ON 4 FEB 76 TO IMPLEMENT SSW(71)
4.  C      VERSION OF 26 OCT 75 TO USE GABS FORMAT
5.  C      VERSION OF 13 SEPT 75 TO USE GCAN
6.  C      INPUT JFMT NUMBER
7.  C      PROGRAM CHARTG, ADAPTED FROM CHART3
8.  C      PROGRAM CHART3, ADAPTED FROM CHART AND CHART2 ON 28 JUNE 1971
9.  C
10. C      VERSION OF MAY 30 1973
11. C      VERSION OF 24 JANUARY READS ONLY GSUM
12. C      VERSION OF 20 DEC 1971, ADD OPTION TO PLOT ONLY GRID
13. C      VERSION OF 8 OCT 1971, CHANGE LOCATION OF PLOT OF TIME
14. C      VERSION OF 5 OCT 1971, ADDING COUNTER TO POINTS PLOTTED
15. C      VERSION OF 23 SEPT 1971, PUTTING LOCAL VARIABLES IN COMMON
16. C      VERSION OF 2 SEPT 1971, CHANGING TO SPOT2
17. C
18. C      PROGRAM PRODUCES MERCATOR CHARTS FROM GSUM DATA
19. C      WITH VALUES PLOTTED BY POINTS OR ALONG TRACK
20. C      WITH TIME ANNOTATION OPTION, DATE IS WRITTEN AT EACH CHANGE OF DAY.
21. C
22. C
23. C
24. C
25. C      SUBROUTINES USED: GRID2, OLINE, WHR, STAT, ISW, SPOT2, CALCOMP ROUTINES, RETRY,
26. C      VETBY, TODAY, ARLIM, ENDIO, EVIL, SHTV, DMTOR, FIND,
27. C      ANBV2, POSTAP
28. C
29. C
30. C1 PLOT LABEL
31. C2 SENSE SWITCH OPTIONS
32. C3 TIME INTERVAL - READS START AND END DATE. IF TAPE INPUT, TAPE CAN BE PRE-
33. C   POSITIONED BY SPECIFYING RECORDS TO BE SKIPPED. IF NOT END DATE SPECIFIED
34. C   NO TEST MADE FOR END DATE.
35. C4 PLOT FORMATS 1. SCALE IN INCHES PER DEGREE LONGITUDE 2. CONNECT PLOTTED
36. C   POINTS 3. PLOT EVERY NTH POINT 4. PLOTTING EVERY NTH GRID LINE
37. C   5. PLOT NUMBER 6. VALUE TO BE ANNOTED 7. ANNOTATE EVERY NTH POINT 8. FORMAT
38. C5 PLOT FORMATS CONTINUED 1. =1 2. CHARACTER HEIGHT (*0.07 INCH) 3. INTEGER
39. C   OR NON-INTEGER CHART BOUNDS 4. DIGITS AFTER DECIMAL POINT PLOTTED
40. C6 PLOT BOUNDARIES - TOP, BOTTOM, LEFT, AND RIGHT EDGES - IN DEGREES AND MINUTES
41. C
42. C
43. C      SSW(0) UP TO NOT READ ANY INPUT DATA, PLOT ONLY GRID
44. C      SSW(1) UP TO DELETE DRAWING NDEG LINES
45. C      SSW(2) UP IF NEXT PLOT WILL BE ON THE SAME GRID AS THIS PLOT.
46. C      SSW(3) UP TO ANNOTATE ONLY AT CHANGE OF DAY
47. C      SSW(4) =0 FOR NO MARK AT DATA POINT
48. C      =1 FOR PLOTTING A CIRCLE AROUND DATA POINT
49. C      =2 FOR PLOTTING A DOT AT DATA POINT
50. C      SSW(5) =0 TO MAKE DEGREE ANNOTATIONS INSIDE GRID (CHARACTER SIZE 0.07 INCH)
51. C      =1 TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.21 INCH)
52. C      =2 TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.35 INCH)
53. C      SSW(6) FOR MULTIPLT RUNS, UP WILL PUT AN EOF BETWEEN PLOTS. USEFUL TO PDP-5
54. C      OPERATOR IN THE EVENT OF MECHANICAL MALFUNCTION OF PEN.
55. C      SSW(7) UP TO ANNOTATE ON LEFT SIDE OF TRACK
56. C      DOWN TO ANNOTATE ON RIGHT SIDE OF TRACK
57. C      SSW(8) UP TO SUPPRESS PLOTTING OF GRID
58. C      SSW(9) UP TO LIST DATE AND TIME OF DATA BUT OF CHART BOUNDARIES
59. C      SSW(10) UP TO CALL SUBROUTINE MBUNT WHICH READS IN SERIAL NUMBER OF INPUT TAPE

```

```

60. C      USED FOR PLOTS HAVING TWO OR MORE INPUT TAPES
61. C SSW(11) UP TO ANNOTATE POINTS ALTERNATELY ON LEFT AND RIGHT SIDE OF TRACK
62. C SSW(12) UP TO LIST DATE OF DATA JUST READ FOR IDENTIFICATION
63. C SSW(13) UP IF TWO OR MORE PLOTS ARE BEING MADE FROM THE SAME TAPE AND THIS IS
64. C      NOT THE LAST PLOT. TAPE WILL BE POSITIONED AT BEGINNING OF CURRENT
65. C      FILE EVEN IF END-OF-FILE PASSED.
66. C SSW(18) =0 TO MAKE ANNOTATION AT RIGHT ANGLES TO INCREMENTAL TRACK (ANDV2)
67. C      =1 TO MAKE ANNOTATIONS HORIZONTALLY
68. C      =2 TO MAKE ANNOTATIONS VERTICALLY
69. C      =3 TO INVERT ANNOTATIONS FOR HEADINGS 180 TO 269
70. C SSW(19) =0 FOR EARTH MERIDIONAL PARTS FROM BOWDITCH
71. C      =1 FOR MERIDIONAL PARTS FOR SPHERICAL PLANET
72. C SSW(71)=N TO ANNOTATE EVERY N HOURS
73. C
74. C
75. C      DIMENSION IBUF(1000),LABELP(20),NOW(4),IEND(4)
76. C      COMMON IBUF, LABELP, NOW, IIN, IIBUT,
77. C      1 MFILE, IND, IFLAG, I, NZERO, NWON,
78. C      2 ZERO, INZ, ITAPE, NAME, PLACE, INIT,
79. C      3 XOLD, YOLD, JODA, JOMB, JOYR, JOHM,
80. C      4 NY, NZ, NW, IEGD, IGAP, IAGAP
81. C      COMMON NPTS, ISTDA, ISTMB, ISTYR, ISTHM, IENDA,
82. C      1 IENMB, IENYR, IENHM, ISKP, ISFIL, IBCKUP,
83. C      2 RADEG, DEGRA, SINCH, ITRK, LCNT, NDEG,
84. C      3 NUMPL, NX, NPTA, JFMT, MFILE, NSKIP,
85. C      4 NSTOP, MSTOP, KPT, KHT, ICTYP, IDEC
86. C      COMMON ZZ, ZHT, IDATA, DATA, RLAT, RLONG,
87. C      1 KOGHM, KDEG2, IDEG2, FDEG2, RDEG2, RTOP,
88. C      2 ITOP, RBOT, IBOT, RLEFT, ILEFT, RRIGHT,
89. C      3 IRIGHT, SLTK, SLGK, SMP, FBOT, FTOP,
90. C      4 FLEFT, FRIGHT, SLAT, SLONG, BOTMP, XX
91. C      COMMON YY, A, B, IP, DATAZ, DATAW,
92. C      1 DATAX, JDA, JMB, JYR, JHM, RLTS,
93. C      2 RLGB, DATAY, KGDA, KGMb, KGYR, KGHM,
94. C      3 IBACK, XINCR, INDK
95. C
96. C      IIN = 105
97. C      IIBUT = 108
98. C      MFILE=0
99. C      PRINT DATE AND TIME OF JOB ON HEADING
100. C      CALL TODAY(NOW)
101. C      WRITE(IIBUT,13) NOW
102. C      13 FORMAT(1X,4A4)
103. C      OUTPUT 'PROGRAM CHARTG VERSION OF 4 FEB 76'
104. C
105. C      CALL SETSKP(IND)
106. C      10 CALL PLOTS (IBUF, -1000)
107. C      MOVE PEN IN FROM EDGE AND ALONG PAPER SO THAT IF OUTSIDE GRID LETTERING WILL
108. C      NOT RUN OFF EDGE.
109. C      CALL PLOT(0.5,0.5,-3)
110. C      CALL STAT
111. C      79 IFLAG=0
112. C      CARD 1 PLOT LABEL (20A4)
113. C      PUT LABEL ON LEFT MARGIN OF PLOT. SHIP, CRUISE, DATE, AND AREA. 80 CHAR.
114. C      HEAD(IIN,6,END=91,ERR=91) LABELP(I), I=1,20
115. C      IF (ISW(5).EQ.1) PLACE=-1.35
116. C      6 FORMAT(20A4)
117. C      WRITE (IIBUT,7) LABELP(I), I=1,20
118. C      7 FORMAT(//,PLOT LABEL: ',20A4)
119. C      NZERO=0

```

```

120.      NWBN=1
121.      ZERO=0.0
122. CARD 2  SENSE SWITCH OPTIONS - SW1 TO SW79 IN COLUMNS 1 TO 79, SW0 IN COLUMN 80
123.      INZ = ISW(=2)
124.      IF (ISW(19).EQ.0) OUTPUT 'MERIDIONAL PARTS FOR EARTH FROM BOWDITCH'
125.      IF (ISW(19).EQ.1) OUTPUT 'MERIDIONAL PARTS FOR A SPHERICAL PLANET'
126.      ITAPE=1
127.      IF ((ISW(10).EQ.1)) READ (IIN,5765) NAME ,CALL MOUNT (ITAPE,NAME)
128.      *WRITE (IIBUT,5784) NAME
129.      5784  FORMAT ( 'USING INPUT TAPE NUMBER' 1X,A4)
130.      5765  FORMAT (A4)
131.      C      POSITION LABEL DEPENDING WHETHER ANNOTATION INSIDE OR OUTSIDE GRID
132.      PLACE=0.80
133.      CALL SYMBOL (PLACE,0.5,0.14,LABELP,90.0,80)
134.      C      ANNOTATE DATE CHART MADE IN LOWER LEFT CORNER OF PLOT
135.      CALL SYMBOL (PLACE,0.48,0.07,NOW,0.0,16)
136.      CALL PLOT (0.0,0.0,3)
137.      C
138.      C
139. CARD 3  TIME INTERVAL OF DATA, AND IF TAPE INPUT - POSITION ON TAPE.
140.      5  INIT=NWBN
141.      XOLD=ZERO
142.      YOLD=ZERO
143.      JSDA=NZERO
144.      JMB=NZERO
145.      JYR=NZERO
146.      JHM=NZERO
147.      NY=NWBN
148.      NZ=NWBN
149.      NW=NWBN
150.      IEND=NZERO
151.      IGAP=NZERO
152.      IAGAP=NZERO
153.      NPTS=NZERO
154.      IPT=NZERO
155.      RADEG=57.29578
156.      DEGRA=1.745329E-2
157.      READ (IIN,2) ISTDA,ISTMO,ISTYR,ISTHM,IENDA,IENMO,IENYR,IFNHM,ISKP,
158.      1 ISFIL,IBCKUP
159.      2  FORMAT (3I2,I4,5X,3I2,I4,5X,3I5)
160.      WRITE (IIBUT,6365) ISTDA,ISTMO,ISTYR,ISTHM,IENDA,IENMO,IENYR,
161.      1 IFNHM,ISKP,ISFIL
162.      6365  FORMAT (1,START DATE 1,3I2,1X,I4,5X,END DATE 1,3I2,1X,I4,5X,RECORD
163.      1S SKIPPED TO START OF INTERVAL (ISKP): 1,I4,2X,FILES SKIPPED TO
164.      2START OF INTERVAL (ISFIL): 1,I4)
165.      C  SPACING FILES ON MULTIFILE INPUT TAPES
166.      IF (ISFIL.EQ.0) GO TO 11
167.      CALL SKPFIL (ITAPE,ISFIL,'FWD')
168.      GO TO (997,11,11,997,997,997) IND
169.      C  SPACING RECORDS ON INPUT TAPE
170.      11  IF (ISKP.EQ.0) GO TO 8
171. CARD 4  PLOT FORMAT PARAMETERS
172.      8  READ (IIN,3) SINCH,ITRK,LCNT,NDEG,NUMPL,NPTA,JFMT,NX,NFILE,IBLK
173.      3  FORMAT (F10.0,3I5,A5,5I5)
174.      WRITE (IIBUT,6498) SINCH,ITRK,LCNT,NDEG,NUMPL,NPTA,JFMT,NX,IBLK
175.      6498  FORMAT ('CHART SCALE (SINCH) 1,F7.3, INCHES PER DEGREE LONGITUDE'
176.      1,4X,'TRACK POINTS CONNECTED (ITRK): 1,I2,4X,'PLOTING EVERY NTH PSI
177.      2NT (LCNT): 1,I2,2X,'PLOTING EVERY NTH DEGREE LINE (NDEG): 1,I2,4X,
178.      3'PLOT NUMBER (NUMPL): 1,A5,4X,'ANNOTATE EVERY NTH PLOTTED POINT (NP
179.      4TA): 1,I2,2X,'DATA FORMAT (JFMT): 1,I2,4X,'VALUE ANNOTATED (NX): 1,

```

```

180.      * I2,/,2X,'BLOCKING FACTOR OF INPUT TAPE (IBLK):',I4)
181.      WRITE(IIOUT,6493) NFILE
182.      6493 FORMAT(' NUMBER OF FILES BEING BUTPUTTED ON SAME GRID (NFILE):')
183.      C JFMT 3 ... GSUM DATA
184.      C
185.      NSKIP=LCNT-1
186.      NSTOP=LCNT
187.      MSTOP=NPTA
188.      CARD 5 PLOT FORMAT PARAMETERS
189.      READ(IIN,4)KPT,KHT,ICTYP,IDEC
190.      4 FORMAT(4I5)
191.      WRITE(IIOUT,6499) KPT,KHT,ICTYP,IDEC
192.      6499 FORMAT('MAGNIFICATION FACTOR (KPT)='/,I2,4X,'CHARACTER HEIGHT',
193.      1 INCH (KHT)='/,I2,4X,'NON-INTEGER OR INTEGER CHART BRUNDS (ICTYP):',
194.      2,I2,2X,'DECIMAL POINT IN ANNOTATION (IDEC):',I2)
195.      C
196.      C IDEC=VARIABLE FOR DECIMAL POINT IN ANGV ANOTATION
197.      C IDEC=N FOR N DIGITS TO RIGHT OF DECIMAL PT.
198.      C      =0 DECIMAL PT ONLY
199.      C      -1 SUPPRESS DECIMAL POINT
200.      ZZ = KPT
201.      SINCH = SINCH * ZZ
202.      ZHT=KPT*KHT
203.      CARD 6 PLOT BOUNDARIES (CARDS 6 TO 9 IF NON-INTEGER BRUNDS)
204.      C ITOP, IBOT, ILEFT, IRIGHT READ IN BY RETBY
205.      WRITE(IIOUT,6364)
206.      6364 FORMAT('CHART BOUNDARIES:',20X,'EAST AND NORTH POSITIVE')
207.      IF(ICTYP)205,205,210
208.      210 CALL RETBY
209.      A( IDATA,IEOD,IIN,IIOUT,ITAPE,NUMPL,DATA,RLAT,RLONG,K9GHM,IAGAP,LCN
210.      BT,RADEG,DEGRA,KDEG2,IDEG2,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,RLEFT,
211.      C ILEFT,RRIGT,IRIGT,SLTK,SLGK,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,NDEG,
212.      D SLAT,SLONG,BOTMP,XX,YY,INIT,XOLD,YOLD)
213.      GO TO 230
214.      C DEGREES AND MINUTES FOR RTOP, RBOT, RLEFT, RRIGHT READ IN BY
215.      C ARLIM AS CALLED BY VETBY
216.      205 CALL VETBY
217.      A( IDATA,IEOD,IIN,IIOUT,ITAPE,NUMPL,DATA,RLAT,RLONG,K9GHM,IAGAP,LCN
218.      BT,RADEG,DEGRA,KDEG2,IDEG2,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,RLEFT,
219.      C ILEFT,RRIGT,IRIGT,SLTK,SLGK,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,NDEG,
220.      D SLAT,SLONG,BOTMP,XX,YY,INIT,XOLD,YOLD)
221.      230 A = ABS(RBOT)/2.0*(45.0*DEGRA)
222.      B = (ALOG(SIN(A)/COS(A))) * 0.4342945
223.      BOTMP=7.915704E+03 *B-(23.26893*SIN(ABS(RBOT
224.      1 )))-(0.0525*(SIN(ABS(RBOT))))**3)
225.      IF(RBOT) 20,30,30
226.      20 BOTMP = -BOTMP
227.      30 IF(ICTYP)22,22,32
228.      C SUBROUTINE BLINE FOR GRID ENCLOSED BY NON-INTEGER DEGREES
229.      22 CALL BLINE(ZZ,ZHT,
230.      A IDATA,IEOD,IIN,IIOUT,ITAPE,NUMPL,DATA,RLAT,RLONG,K9GHM,IAGAP,LCN
231.      BT,RADEG,DEGRA,KDEG2,IDEG2,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,RLEFT,
232.      C ILEFT,RRIGT,IRIGT,SLTK,SLGK,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,NDEG,
233.      D SLAT,SLONG,BOTMP,XX,YY,INIT,XOLD,YOLD)
234.      C
235.      C CHECKING IF ONLY DRAWING OF GRID WANTED
236.      C
237.      IF(IISW(0))310/310/91
238.      C
239.      C SUBROUTINE GRID2 FOR GRID ENCLOSED BY WHOLE DEGREES

```

```

240.      32 CALL GRID2(ZZ,ZHT,NUMPL,DEGRA,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,
241.      1 RLEFT,ILEFT,RRIGHT,IRIGHT,SINCH,SMP,FBOT,FTOP,LEFT,FRIGHT,NDEG,
242.      2 SLAT,SLONG,BOTMP)
243.      C
244.      C      CHECKING IF ONLY DRAWING OF GRID WANTED
245.      C
246.      IF(ISW(0))310,310,91
247.      C
248.      C END OF INITIALIZATION, BEGIN PLOTTING PRINTS
249.      C
250.      C      CHECKING IF TRACK POINTS SHOULD BE CONNECTED
251.      35 IF(ITRK)40,40,45
252.      40 IP = 3
253.      GO TO 70
254.      45 IF(INIT) 55,55,50
255.      50 IP = 3
256.      GO TO 70
257.      55 IF(IGAP) 65,65,60
258.      60 IP = 3
259.      GO TO 70
260.      65 IP = 2
261.      C      PLOT DATA POINT
262.      70 SLAT = RLAT
263.      SLONG = RLONG
264.      CALL WHR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMP,SLAT,SLONG,BOTMP,XX,YY)
265.      CALL PLOT(XX,YY,IP)
266.      IPCT=IPCT+1
267.      IF(ISW(16))876,876,710
268.      710 CALL ANBV3(XX,YY,DATAZ,DATAW)
269.      GO TO 72
270.      C      CHECK FOR TYPE OF DATA POINT ANNOTATION MADE IN SPOT2
271.      876 CONTINUE
272.      73 CALL SPOT2(XX,YY)
273.      IF(ISW(18),EQ,5) GO TO 72
274.      C      DATA POINT PLOTTED
275.      74 DATA=DATAZ
276.      IF(NX)71,72,71
277.      71 MSTOP=MSTOP + NWON
278.      IF(MSTOP)871,871,72
279.      871 CONTINUE
280.      C
281.      C      TEST FOR TIME INTERVAL ANNOTATION
282.      C
283.      IF(ISW(71)) 872,878,872
284.      872 IF(JDA=JBDA)875,873,875
285.      873 IF(FLBAT(JHM/100)=FLBAT(JBHM/100)) 874,72,874
286.      874 ITCT=ITCT+1
287.      IF(ITCT.LT.ISW(71)) GO TO 72
288.      875 ITCT=0
289.      878 CONTINUE
290.      CALL ANBV6(ZZ,ZHT,NX,JDA,JMB,JYR,JHM,JBDA,XX,YY,INIT,DATA,IDEC)
291.      MSTOP=NPTA
292.      72 INIT = 0
293.      RLTO = RLAT
294.      RLGO = RLONG
295.      JBDA=JDA
296.      JMB=JMB
297.      JYR=JYR
298.      JBHM=JHM
299.      NPTS=NPTS+1

```

```

300. 310 CONTINUE
301. C          GSUM DATA
302.          IF(JFMT.EQ.3)
303.          *CALL GETGS(ITAPE,NX,DATAX,JFMT,RTOP,RBOT,RLEFT,RRIGT,
304.          1 RLAT,RLONG,JDA,JMO,JYR,JHM,IEBD,IBLK)
305. C          GCON DATA
306.          IF(JFMT.EQ.13)
307.          * CALL GETGC(ITAPE,NX,DATAX,NY,DATAY,NZ,DATZ,
308.          * RTOP,RBOT,RLEFT,RRIGT,RLAT,RLONG,IEBD)
309. C          GABS DATA
310.          IF(JFMT.EQ.14)
311.          * CALL GETGA(ITAPE,NX,DATAX,NY,DATAY,NZ,DATZ,
312.          * RTOP,RBOT,RLEFT,RRIGT,RLAT,RLONG,IEBD)
313. 78 CONTINUE
314.          IF (IEBD=1) 85,1212,1212
315. 1212 NFILE=NFILE+1
316.          IF(NFILE.LE.0) GO TO 12
317.          IEBD=0
318.          MFILE=MFILE+1
319.          OUTPUT 'PLOTTING CONTINUES FROM NEXT FILE'
320.          GO TO 85
321. C
322. C END OF DATA - CHECK SENSE SWITCHES FOR NEXT STEP
323. C
324. C CHECKING IF ANOTHER PLOT FOLLOWS OR TO EXIT
325. 12 IF(ISW(2).EQ.0.AND.ISW(13).EQ.0.AND.IBACKUP.EQ.0) GO TO 91
326.          IF(ISW(2).EQ.1) CALL PLBT(0.0,0.0,3)
327. C SENSE SWITCH 13 CHECKS WHETHER ANOTHER PLOT IS TO BE MADE FROM SAME FILE.
328.          IF(ISW(13)) 80,81,80
329. 80 IBACK=0
330.          IF(IEBD.EQ.1) IBACK=1
331.          IF(MFILE.NE.0) IBACK=MFILE+1
332.          CALL SKPFIL(ITAPE,IBACK,IREV)
333. C SENSE SWITCH 14 CHECKS WHETHER ANOTHER INPUT TAPE IS TO BE MOUNTED.
334. 81 IF (ISW(14)) 83,83,87
335. 87 WRITE(11OUT,6496)
336. 6496 FORMAT(/,CHANGE TO NEXT INPUT TAPE,/)
337.          CALL MCVBL(1)
338. C SEQUENTIAL PLOTS FROM SAME FILE WITH (IBACKUP) OVERLAP
339. 83 IF(IBACKUP.EQ.0) GO TO 96
340.          IBACKUP=IBACKUP+1
341.          IF(IEBD.EQ.0) CALL SKPREC(ITAPE,IBACKUP,IREV)
342.          IF(IEBD.EQ.1) CALL SKPFIL(ITAPE,1,IREV)
343. C ESTABLISH ORIGIN OF NEW PLOT
344. 96 IF (ISW(2).EQ.1) GO TO 95
345.          XINCRE=ABS(RDEG2)*RADEG*SINCH+4.00
346.          CALL PLBT(XINCRE,0.0,3)
347. 95 IF(ISW(6).EQ.1) CALL PLBT(-1.00,-1.00,999), GO TO 94
348. C PUT RUN AND DATA CARDS FOLLOWING LAST DATA DECK TO REINITIALIZE PROGRAM.
349.          GO TO 79
350. C CLOSE PLOT TAPE AND END OF JOB LABEL FOR PDP-5 OPERATOR
351.          DATA (IEND(I),I=1,4) /'END OF CHART JOB'/
352. 91 XINCRE=ABS(RDEG2)*RADEG*SINCH+1.00
353. C COMPLETE BOTTOM AND RIGHT SIDES OF FIDUCIAL HALF-INCH SQUARE DRAWN IN GRID2
354.          CALL PLBT(XINCRE,-0.5,-3)
355.          CALL PLBT(0.5,0.0,-2)
356.          CALL PLBT(0.0,0.5,-2)
357.          CALL PLBT(2.5,-0.5,-3)
358.          CALL SYMBL(0.0,0.0,0.42,IEND,90.0,16)
359.          CALL PLBT(4.0,0.0,999)

```

```

360.      94  IF(ISW(10).EQ.1) CALL MREL(1)
361.      WRITE(IIOUT,97)IPCT
362.      97  FORMAT('  NUMBER POINTS PLOTTED = ',I8)
363.      CALL EXIT
364.      C    SKIPPING POINTS, IF EVERY POINT NOT TO BE PLOTTED.
365.      85  IF(NSKIP) 186,186,185
366.      185  CALL SKPREC(ITAPE,NSKIP)
367.      GO TO (999,186,12,999,12,999)IND
368.      186  CONTINUE
369.      IF(JFMT.EQ.13) GO TO 852
370.      IF(JFMT.EQ.14) GO TO 852
371.      C    CHECK IF DATE IS WITHIN SPECIFIED INTERVAL
372.      C    IFLAG IS A FLAG TO ALLOW SKIPPING THE FIRST CALL TO FIND
373.      C    IF WE HAVE ALREADY FOUND THE STARTING DATE
374.      IF(IFLAG.NE.0) GO TO 82
375.      CALL FIND(ISTDA,ISTMB,ISTYR,ISTHM,JDA,JMB,JYR,JHM,INDK)
376.      IF(INDK.EQ.-1) GO TO 310
377.      IFLAG=1
378.      82  CONTINUE
379.      IF(IENYR.EQ.0) GO TO 851
380.      CALL FIND(IENDA,IENMB,IENYR,IENHM,JDA,JMB,JYR,JHM,INDK)
381.      IF(INDK.EQ.1) GO TO 995
382.      851  CONTINUE
383.      852  CONTINUE
384.      C    CHECKING IF DATA WITHIN CHART BOUNDRIES
385.      NSTOP=LCNT
386.      IF(RTOP=RLAT) 100,100,86
387.      86  IF(RLAT=RBOT) 100,88,88
388.      88  IF(RLONG=RLEFT) 100,90,90
389.      90  IF(RRIGT=RLONG) 100,100,92
390.      C    DATA WITHIN BOUNDS
391.      92  GO TO 35
392.      C    DATA OUTSIDE OF BOUNDS
393.      C    WRITING OUT DATE IF DATA POINT IS OUT OF BOUNDS
394.      100  INIT=1
395.      IF(ISW(9))410,310,410
396.      410  WRITE(IIOUT,420)JDA,JMB,JYR,JHM
397.      420  FORMAT('000 ',3I2,1X,I4)
398.      GO TO 310
399.      995  WRITE(IIOUT,996) JDA,JMB,JYR,JHM
400.      996  FORMAT('END DATE PASSED',2X,3I2,1X,I4)
401.      GO TO 12
402.      C    ERROR MESSAGES IF MISTAKE IN TAPE FILE OR RECORD SPACING.
403.      999  WRITE(IIOUT,998) IND
404.      998  FORMAT('ERROR IN SKPREC, IND=',I2)
405.      GO TO 91
406.      997  WRITE(IIOUT,994) IND
407.      994  FORMAT('ERROR IN SKPFIL, IND=',I2)
408.      CALL EXIT
409.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	00457 C	1	ABS	R	SPRGG	INTRIN	1	ALOG	R	SPRGG	INTRIN	1	BACKUP	I	SCALR	00405 C	1
ANOVG	R	SPRGG	EXTERN	1	ANOV3	R	SPRGG	EXTERN	1	B	R	SCALR	00458 C	1	IBUF	I	SCALR	00425 C	1
B0TMP	R	SCALR	00454 C	1	CMS	R	SPRGG	INTRIN	1	DATA	R	SCALR	00438 C	1	IBUF	I	SCALR	00425 C	1
DATAM	R	SCALR	00458 C	1	DATA	R	SCALR	0045C C	1	DATA	R	SCALR	00463 C	1	IBUF	I	SCALR	00425 C	1
DATAX	R	SCALR	0045A C	1	DEGRA	R	SCALR	00427 C	1	EXIT	R	SPRGG	EXTERN	1	IBUF	I	SCALR	00425 C	1
F89T	R	SCALR	00450 C	1	FDEG2	R	SCALR	00441 C	1	FIND	R	SPRGG	EXTERN	1	IBUF	I	SCALR	00425 C	1
FLEET	R	SCALR	00450 C	1	FLBAT	R	SPRGG	EXTERN	1	FRIGT	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
F79P	R	SCALR	00450 C	1	GETGA	R	SPRGG	EXTERN	1	GETGC	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
GETGS	R	SCALR	00450 C	1	GHID2	R	SPRGG	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
IAGAP	R	SCALR	00419 C	1	IBACK	R	SPRGG	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
IBLK	R	SCALR	00005 V	1	IBOT	R	SCALR	00468 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
ICTYP	R	SCALR	00436 C	1	IDATA	R	SCALR	0043A C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
IDEG2	R	SCALR	00440 C	1	IEND	R	SCALR	00000 V	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
IENHM	R	SCALR	00422 C	1	IENM0	R	SCALR	00420 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
IE9D	R	SCALR	00417 C	1	IFLAG	R	SCALR	00404 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
ITN	R	SCALR	00400 C	1	IIBUT	R	SCALR	00401 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
IND	R	SCALR	00403 C	1	INDK	R	SCALR	0046A C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
INZ	R	SCALR	00409 C	1	IP	R	SCALR	00459 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
IRIGT	R	SCALR	0044A C	1	ISFIL	R	SCALR	00424 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
ISTDA	R	SCALR	0041B C	1	ISTHM	R	SCALR	0041E C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
ISTYR	R	SCALR	0041D C	1	ISW	R	SCALR	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
ITCT	R	SCALR	00006 V	1	IT8P	R	SCALR	00444 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
JDA	R	SCALR	0045D C	1	JMT	R	SCALR	0042F C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
J8	R	SCALR	0045E C	1	JDA	R	SCALR	00410 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
J8M8	R	SCALR	00411 C	1	JYR	R	SCALR	00412 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
KDEG2	R	SCALR	0043F C	1	KDA	R	SCALR	00464 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
K8M8	R	SCALR	00465 C	1	KGR	R	SCALR	00466 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
LCNT	R	SCALR	0043E C	1	KPT	R	SCALR	00434 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
MCNT	R	SCALR	0042A C	1	MCV8L	R	SPRGG	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
MCNT	R	SCALR	EXTERN	1	MREL	R	SPRGG	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
NAME	R	SCALR	00408 C	1	NDEG	R	SCALR	0042B C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
NSW	R	SCALR	00431 C	1	NPTA	R	SCALR	0042E C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
NSKIP	R	SCALR	00431 C	1	NST8P	R	SCALR	00432 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
NH	R	SCALR	00416 C	1	NW8N	R	SCALR	00407 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
NY	R	SCALR	00414 C	1	NZ	R	SCALR	00415 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
PLINE	R	SCALR	EXTERN	1	PLACE	R	SCALR	0040C C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
PLSTS	R	SCALR	EXTERN	1	RDEG	R	SCALR	00426 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
RDEG2	R	SCALR	00442 C	1	RETBY	R	SCALR	00462 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
RLEFT	R	SCALR	00447 C	1	RUG8	R	SCALR	00462 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
RLT8	R	SCALR	00461 C	1	RIGT	R	SCALR	00449 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
SETSKP	R	SCALR	EXTERN	1	SIN	R	SCALR	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
SKPFIL	R	SCALR	EXTERN	1	SKPREC	R	SCALR	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
SLGK	R	SCALR	0044C C	1	SL8NG	R	SCALR	00453 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
SP8K	R	SCALR	0044D C	1	SP8T2	R	SCALR	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
SYMB8L	R	SCALR	EXTERN	1	T8DAY	R	SCALR	EXTERN	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
XX	R	SCALR	EXTERN	1	XINCRE	R	SCALR	00449 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
ZERR	R	SCALR	00455 C	1	Y8LD	R	SCALR	00469 C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1
			00408 C	1	ZHT	R	SCALR	0046F C	1	IBUF	R	SCALR	00451 C	1	IBUF	I	SCALR	00425 C	1

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
2	000FA	7	00037	5	000C9	11	0014B	13	00140
8	0014D	20	002C4	12	003FD	32	002FE	40	00120
22	002C9	45	00323	35	0031E				

50	00325	55	00328	60	0032A	65	0032D	70	0032F	71	0035F
72	0038E	73	00352	74	0035B	78	003D8	79	0032F	80	0040B
81	0041C	82	004C6	83	00431	85	0049D	86	004DB	87	00421
88	004DE	90	004E1	91	0045E	92	004E4	94	00484	95	00452
96	00443	97	00491	100	004E5	185	0049F	186	004AE	205	00275
210	00247	230	002A2	310	0039D	410	004EC	420	004F4	710	0034B
851	004D6	352	004D6	871	00364	872	00369	873	0036C	874	00375
875	0037B	876	00352	878	0037D	994	00520	995	004FB	996	00503
997	0051B	998	00512	999	0050D	1212	003D3	5765	000BQ	5784	000A6
6364	00237	6365	00110	6493	001D2	6496	00425	6498	0016E	6499	001FC

## LOCAL VARIABLES (7 WORDS):

00000	IEND	00004	IPCT	00005	IBLK	00006	ITCT
-------	------	-------	------	-------	------	-------	------

## BLANK COMMON (1131 WORDS):

00000	IBUF	003E8	LARELP	003FC	N8W	00400	IIN	00401	IIOUT	00402	MFILE
00403	IND	00404	IFLAG	00405	I	00406	NZER0	00407	NMON	00408	ZER0
00409	INZ	0040A	ITAPE	0040B	NAME	0040C	PLACE	0040D	INIT	0040E	XOLD
0040F	YOLD	00410	J0DA	00411	J0M0	00412	J0YR	00413	J0HM	00414	NY
00415	NZ	00416	NW	00417	IE0D	00418	IGAP	00419	IAGAP	0041A	NPTS
0041B	ISTDA	0041C	ISTM0	0041D	ISTYR	0041E	ISTHM	0041F	IENDA	00420	IENM0
00421	IENYR	00422	IENHM	00423	ISKP	00424	ISFIL	00425	IBCKUP	00426	RADEG
00427	DEGRA	00428	SINCH	00429	ITRK	0042A	LCNT	0042B	NDEG	0042C	NUMPL
0042D	NX	0042E	NPTA	0042F	JFMT	00430	NFILE	00431	NSKIP	00432	NST0P
00433	MST0P	00434	KPT	00435	KHT	00436	ICTYP	00437	IDEC	00438	ZZ
00439	ZHT	0043A	IDATA	0043B	DATA	0043C	RLAT	0043D	RL0NG	0043E	K0G0M
0043F	KDEG2	00440	IDEG2	00441	FDEG2	00442	RDEGP	00443	RT0P	00444	IT0P
00445	R0T	00446	IRAT	00447	RLEFT	00448	RLEFT	00449	RRIGT	0044A	IRIGT
00448	SLTK	0044C	SLGK	0044D	SMP	0044E	F0BT	0044F	FT0P	00450	FLEFT
00451	FRIGT	00452	SLAT	00453	IP	00454	R0TMP	00455	XX	00456	YY
00457	A	00458	B	00459	IP	0045A	DATZ	0045B	DATAM	0045C	DATAX
0045D	JDA	0045E	JM0	0045F	JYR	00460	JHM	00461	RLT0	00462	RLG0
00463	DATAY	00464	KGDA	00465	KGM0	00466	KGYR	00467	KGMH	00468	IBACK
00469	XINCRC	0046A	INDK								

## INTRINSIC SUBPROGRAMS USED:

ABS	AL0G	COS	FL0AT	SIN
-----	------	-----	-------	-----

## EXTERNAL SUBPROGRAMS REQUIRED:

AN0V0	AN0V3	EXIT	GETGA	GFTGC	GETGS	GRID2
ISW	MCVPL	MAUNT	BLINE	PL0T	PL0TS	RETBY
SETSKP	SKPFL	SKPREC	STAT	SYMB0L	T0DAY	VETBY
W0R	F:101	F:102	F:104	F:105	F:106	F:108
M:09	M:0C	9AL0G	9ACDRE0	9ACDWRIT	9C0S	9ENDIAL
9INITIAL	9IBDATA	9IBLUSA	9PRINT	9SIN	9SY0P	

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS	
GENERATED CODE:	1324	0052C	(NO MEMORY PROTECTION)
CONSTANTS:	44	0002C	
LOCAL VARIABLES:	7	00007	
TEMPS:	3	00003	
TOTAL PROGRAM:	1378	00562	(PLUS BLANK COMMON)

```

1.      1 FORMAT(1X,'PROGRAM CONV67 VERSION 17 JAN 74')
2.      C      PROGRAM CONV67
3.      C      VERSION 17 JAN 74 TO SPEED THINGS UP BY DECODING FEWER ITEMS
4.      C      PROGRAM TO CONVERT TO 1967 GRAVITY SYSTEM
5.      C      AND NEW GEODETIC REFERENCE
6.      C      ORIGINAL VERSION 1 DEC 73 BY LEE GØVE
7.      DIMENSION IBUFIN(32,50,2), IBLFOT(32,50,2)
8.      DIMENSION IA(35)
9.      WRITE (108,1)
10.     IREC2=2
11.     ITAPE=1
12.     JTAPE=2
13.     KTAPE=108
14.     IFLIP=1
15.     JFLIP=1
16.     KFLIP=1
17.     NFLIP=1
18.     ICNT=0
19.     NIN=50
20.     NOUT=0
21.     IOUTSW=0
22.     DEGRA=1.745329E-2
23.     KI=1,KØ=-2
24.     C
25.     C      BUFFER LOGIC FOR I/P
26.     C
27.     CALL BUFF IN(ITAPE,Ø,IUFIN(1,1,IFLIP),1600)
28. 10 CONTINUE
29.     IF(NIN.LT.50) GØ TO 90
30. 15 CONTINUE
31.     CALL ICHECK(ITAPE,IKEY,NI)
32.     GØ TO (20,50,30,40) IKEY
33. 20 OUTPUT ('WAITING FOR I/P', IEØD=Ø)
34.     GØ TO 15
35. 30 OUTPUT ('END OF FILE ON ITAPE', IEØD=1)
36.     GØ TO 50
37. 40 OUTPUT ('BUFFER IN ERROR', IEØD=1)
38.     GØ TO 999
39. 50 CONTINUE
40.     NIN=Ø
41.     NFLIP=IFLIP
42.     IFLIP=3-IFLIP
43.     IF(IEØD.NE.1) CALL BUFF IN(ITAPE,Ø,IUFIN(1,1,IFLIP),1600)
44.     C
45.     C      INPUT LOGIC
46.     C
47. 90 CONTINUE
48.     NIN=NIN+1
49.     IF (NI.EQ.1600) GØ TO 95
50.     C      GOING TO EOF PROCESSING
51.     NINCHK=NIN*32
52.     IF(NINCHK.GT.NI) GØ TO 999
53. 95 CONTINUE
54.     DECODE(72,1001,IUFIN(1,NIN,NFLIP),ND) IREC1,ISØRC,KGDA,KGMØ,
55. 1     KGYR,KGHM,DLAT,DLØNG,ELEV,K977,ØBSG,IDEØ,FA,BG,TC,IELC
56.     C
57.     C      EDIT LOGIC
58.     C
59.     C

```

```

60. C      CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF
61. C      THE 1967 INTERNATIONAL GRAVITY FORMULA
62. C      AND NEW GEODETIC REFERENCE SYSTEM
63. C
64.      CALL OBG(K977,OBBSG,GOBS,KI)
65.      GOBS*GOBS=14.0
66.      CALL OBG(K977,OBBSG,GOBS,KO)
67.      RLAT=DLAT*DEGRA
68.      DG=3.2*(13.6*(SIN(ABS(RLAT))**2))
69.      IF (FA.LT.990.) FA=FA+DG
70.      IF (BG.LT.990.) BG=BG+DG
71. 100 CONTINUE
72. C
73. C      OUTPUT LOGIC
74. C
75. 300 CONTINUE
76.      NOUT=NOUT+1
77.      DO 301 I=19,32,1
78.      IBUFOT(I,NOUT,JFLIP)=IBUFIN(I,NIN,NFLIP)
79. 301 CONTINUE
80.      ENCODE(72,1001,IBUFOT(1,NOUT,JFLIP),ND)IREC2,ISORC,KGDA,KGM9,
81. 1      KGYR,KGHM,DLAT,DLONG,ELEV,K977,OBBSG,IDEF,FA,BG,TC,IELC
82. 305 CONTINUE
83.      IF(NOUT.LT.50) GO TO 10
84. C
85. C      BUFFER LOGIC FOR O/P
86. C
87. 310 CONTINUE
88.      IF(IOUTSW.NE.1) IOUTSW=1; GO TO 350
89.      JKEY=ICHECK(JTAPE)
90.      GO TO (320,350,330,340) JKEY
91. 320 OUTPUT 'WAITING FOR O/P' ; IEOD=0
92.      GO TO 310
93. 330 OUTPUT 'END OF FILE JTAPE' ; IEOD=1
94.      GO TO 999
95. 340 OUTPUT 'BUFF OUT ERROR' ; IEOD=1
96.      GO TO 999
97. 350 CONTINUE
98.      NOUT=0
99.      KFLIP=JFLIP
100.      JFLIP=3-JFLIP
101.      CALL BUFF OUT(JTAPE,0,IBUFOT(1,1,KFLIP),1600)
102.      GO TO 10
103. C
104. C      END OF JOB
105. C
106. 999 CONTINUE
107. 910 CONTINUE
108.      JKEY=ICHECK(JTAPE)
109.      GO TO (920,950,930,940) JKEY
110. 920 OUTPUT 'WAITING FOR O/P' ; IEOD=0
111.      GO TO 910
112. 930 OUTPUT 'BAD JKEY' ; IEOD=1
113.      GO TO 960
114. 940 OUTPUT 'BUFF OUT ERROR' ; IEOD=1
115.      GO TO 960
116. 950 CONTINUE
117.      JWDS=NOUT*32
118.      CALL BUFF OUT(JTAPE,0,IBUFOT(1,1,JFLIP),JWDS)
119. 960 CONTINUE

```

```
120.      END FILE JTAPE
121.      OUTPUT 'ALL DONE'
122.      C
123.      C      FORMATS
124.      C
125.      98  FORMAT(1X,32A4)
126.      1001 FORMAT(11,I4,3I2,I4,2F9.4,F7.2,I3,F4.2,I5,2F6.1,F4.1,I2)
127.      END
```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
ABS	R	SPRGG	01944	1	BG	R	SCALR	01944	1	BUFFIN	I	SPRGG	01944	1	BUFFIN	I	SPRGG	01944	1
BUFFOUT	R	SPRGG	01945	1	DEGRA	R	SCALR	01945	1	DG	R	SCALR	01945	1	DG	R	SCALR	01945	1
DLAT	R	SCALR	01946	1	DLONG	R	SCALR	01946	1	ELEV	R	SCALR	01946	1	ELEV	R	SCALR	01946	1
PA	R	SCALR	01947	1	GBS	R	SCALR	01947	1	I	I	ARRAY	01947	1	I	I	ARRAY	01947	1
IA	R	ARRAY	01948	35	IBUFIN	R	ARRAY	01948	35	IBUFOT	I	ARRAY	01948	35	IBUFOT	I	ARRAY	01948	35
ICHECK	I	SPRGG	01949	1	ICNT	I	SCALR	01949	1	IDEP	I	SCALR	01949	1	IDEP	I	SCALR	01949	1
IELC	I	SCALR	01950	1	IED	I	SCALR	01950	1	IFLIP	I	SCALR	01950	1	IFLIP	I	SCALR	01950	1
IKEY	I	SCALR	01951	1	IBUTSM	I	SCALR	01951	1	IREC1	I	SCALR	01951	1	IREC1	I	SCALR	01951	1
IREC2	I	SCALR	01952	1	ISRC	I	SCALR	01952	1	ITAPE	I	SCALR	01952	1	ITAPE	I	SCALR	01952	1
JFLIP	I	SCALR	01953	1	JKEY	I	SCALR	01953	1	JTAPE	I	SCALR	01953	1	JTAPE	I	SCALR	01953	1
JNDS	I	SCALR	01954	1	KFLIP	I	SCALR	01954	1	KGDA	I	SCALR	01954	1	KGDA	I	SCALR	01954	1
KGHM	I	SCALR	01955	1	KGO	I	SCALR	01955	1	KGYR	I	SCALR	01955	1	KGYR	I	SCALR	01955	1
KI	I	SCALR	01956	1	KO	I	SCALR	01956	1	KTAPE	I	SCALR	01956	1	KTAPE	I	SCALR	01956	1
K977	I	SCALR	01957	1	ND	I	SCALR	01957	1	NFLIP	I	SCALR	01957	1	NFLIP	I	SCALR	01957	1
NI	I	SCALR	01958	1	NIN	I	SCALR	01958	1	NINCHK	I	SCALR	01958	1	NINCHK	I	SCALR	01958	1
NOUT	I	SCALR	01959	1	NBG	I	SCALR	01959	1	OBEG	R	SCALR	01959	1	OBEG	R	SCALR	01959	1
RLAT	R	SCALR	01960	1	SIN	R	SPRGG	01960	1	TC	R	SCALR	01960	1	TC	R	SCALR	01960	1

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
1	00001	10	00037	15	0003A	20	00048	30	00054
50	0006E	90	00082	95	0008C	98	0018B	100	000CB
301	000DB	305	000F7	310	000FA	320	0010D	330	00119
350	00132	510	00144	920	00151	930	0015D	940	00168
960	00181	999	00144	1001	0018F				

## LOCAL VARIABLES (6477 WORDS):

00000	IBUFIN	00080	IBUFOT	01900	IA	01923	IREC2	01924	ITAPE	01925	JTAPE
01926	KTAPE	01927	IFLIP	01928	JFLIP	01929	KFLIP	0192A	NFLIP	0192B	ICNT
0192C	NIN	0192D	NOUT	0192E	IBUTSM	0192F	DEGRA	01930	KI	01931	KB
01932	IKEY	01933	NI	01934	IEBC	01935	NINCHK	01936	ND	01937	IREC1
01938	ISORC	01939	KGDA	0193A	KGO	0193B	KGYR	0193C	KGHM	0193D	DLAT
0193E	CLONG	0193F	ELEV	0194C	K977	01947	OBEG	01942	IDEP	01943	FA
01944	BG	01945	TC	01946	IELC	01947	GGBS	01948	RLAT	01949	DG
0194A	I			0194C	JNDS						

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

ABS SIN

EXTERNAL SUBPROGRAMS REQUIRED:

BUFFIN	BUFFOUT	ICHECK	9DEC0DE
9ENCODE	9ENDFILE	9END16L	9ST0P

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	415	0019F
CONSTANTS:	7	00007
LOCAL VARIABLES:	6477	0194D
TEMPS:	1	00001
	-----	-----
TOTAL PROGRAM:	6900	C1AF4

```

1. C PROGRAM CR2G
2. C VERSION OF 20 MAR 1973, TO ADD INPUT OF ELEV AND G METER CODES
3. C MODIFICATION BY C. BOWIN
4. C OUTPUT ' *** PGM CR2G OF 20 MAR 1973'
5. C
6. C * MAY 72 -- BY S. ABBOT TO CORRECT OUTPUT LISTING, AND TO CR2G0010
7. C OUTPUT GSUM DATA WITH LAT AND LON IN DECIMAL DEGREES CR2G0020
8. C ALSO TO CLEAN UP THE COMMENTS AND SO ON CR2G0030
9. C * MOD 18 JAN 71/ -S.ABBOT TO CORRECT CR2G0040
10. C ***** REWRITTEN IN A HOPEFUL MANNER BY J. WOLFE JUNE E 1969 CR2G0050
11. C CR2G0060
12. C * INPUT CR2G0070
13. C CR2G0080
14. C * JOB INITIALIZATION CR2G0090
15. C CR2G0100
16. C 1) GRAVITY METER TABLES -- 70 CARDS CR2G0110
17. C 2) SENSE SWITCH CARD -- 8011 (SSW(0) = CC 80) CR2G0120
18. C 3) IGM(1), IGM(2), DRFTCB, LSRC (2A4,2X,F10.5,15) CR2G0130
19. C IGM -- NUMBER AND/OR MODEL OF GRAVITY METER USED CR2G0140
20. C FOR INSTANCE -- L&R G-18 CR2G0150
21. C DRFTCB -- DRIFT CORRECTION FOR GRAVITY METER -- F10.5 CR2G0160
22. C LSRC -- SOURCE CODE FOR GSUM OUTPUT DATA CR2G0170
23. C * NOTE: IF DRFTCB = 0.0, THE ASSUMED VALUE FOR THE GRAVITY METER CR2G0180
24. C DRIFT (ASDFT) WILL BE USED, UNLESS SSW(5) IS ON CR2G0190
25. C * NOTE: IF IGM(1) AND IGM(2) ARE BOTH BLANK, THE GRAVITY METER TYPE CR2G0200
26. C WILL BE SET TO THE DEFAULT TYPE OF 'L&R G-18' CR2G0210
27. C * NOTE: IF LSRC = 0, THE SOURCE CODE WILL BE SET TO THE DEFAULT CR2G0220
28. C VALUE -- LSRC CR2G0230
29. C CR2G0240
30. C * THESE ARE FOLLOWED BY GROUPS OF INDIVIDUAL STATION COUNTER CR2G0250
31. C READING CARDS. EACH GROUP IS HEADED BY 3 CARDS: CR2G0260
32. C 1) THE ABSOLUTE GRAVITY VALUE FOR THE REFERENCE STATION -- CR2G0270
33. C F3.0,F6.2 -- BASEG(1), BASEG(2) CR2G0280
34. C 2) THE CRUSTAL DENSITY IN GM PER CU CM TO BE USED IN CR2G0290
35. C THE CALCULATION OF THE BOUGUER ANOMALY -- F4.2 -- DENSE CR2G0300
36. C 3) THE COUNTER READING CARD FOR THE REFERENCE STATION CR2G0310
37. C CR2G0320
38. C CR2G0330
39. C
40. C FORMAT FOR COUNTER READING DATA IS THAT OF 17 MAY 1966
41. C
42. C * THESE ARE FOLLOWED BY COUNTER READING CARDS FOR THE REST OF THE CR2G0340
43. C MEASUREMENTS THAT ARE TO BE TIED TO THE REFERENCE MEASUREMENT. CR2G0350
44. C CR2G0360
45. C * A COUNTER READING CARD WITH ALL ZEROS EXCEPT FOR THE CR2G0370
46. C YEAR VALUE (CC 9,10) WILL CAUSE THE PGM TO BRANCH TO READ CR2G0380
47. C NEW CARDS FOR BASEG AND DENSE AND THE REF STATION CR2G0390
48. C A CARD WITH ALL ZEROS INCLUDING YEAR WILL GO TO E.O.J. CR2G0400
49. C DO NOT HAVE AN ALL-ZEROS CARD FOLLOWING A CARD WITH ONLY THE YEAR. CR2G0410
50. C CR2G0420
51. C * SENSE SWITCH OPTIONS CR2G0430
52. C CR2G0440
53. C SSW(1) OFF, FOR PRINTED OUTPUT OF COMPUTED VALUES FOR EACH STATION CR2G0450
54. C ON, FOR SUPPRESSION OF PRINTED OUTPUT CR2G0460
55. C SSW(2) OFF, TO PUNCH OUTPUT FOR INPUT TO GRAVITY DESCR. PGM. (GDS) CR2G0470
56. C ON, TO SUPPRESS PUNCHED OUTPUT CR2G0480
57. C SSW(4) OFF, TO OUTPUT GSUM FORMATTED DATA TO 'KTAPE' CR2G0490
58. C ON, TO SUPPRESS GSUM FORMAT OUTPUT CR2G0500
59. C SSW(5) OFF, FOR GRAVITY METER DRIFT CORRECTION CR2G0510
60. C ON, FOR SUPPRESSION OF DRIFT CORRECTION CR2G0520

```

```

60. C
61. C
62. C * VARIABLE DEFINITIONS
63. C
64. C LDAY, MO, LYR, LTIME = DATE AND TIME OF READING -- IF LOCAL TIME
65. C IS USED, KTZ SHOULD ALSO BE ENTERED. IF GMT, KTZ ALWAYS = 0
66. C THIS IS THE TIME AS READ FROM THE COUNTER READING RECORDS.
67. C KGDA, KGMO, KGYR, KGHM = DATE AND TIME OF READING -- IN GMT (OR
68. C LOCAL IF KTZ = 99)
69. C IDA1, IM01, IYR1, ITM1 = DATE AND TIME OF REFERENCE STATION
70. C READING (IN GMT (OR LOCAL IF KTZ = 99))
71. C CRN = GRAVITY METER COUNTER READING (IN UNITS)
72. C ELEV = ELEVATION OF GRAVITY METER ( IN METERS)
73. C KTZ = THE TIME ZONE CORRECTION. IF KTZ = 99, IT INDICATES
74. C THAT KTZ WAS NOT AVAILABLE. THUS CLS AND HONK = 0.0
75. C DATE/TIME MAY BE ENTERED AS GMT, IN WHICH CASE KTZ
76. C WILL ALWAYS BE EQUAL TO ZERO.
77. C IF TIME OF READING IS IN LOCAL TIME BUT TIME ZONE IS NOT
78. C KNOWN, KTZ MAY BE ENTERED AS 99, IN WHICH CASE A TIME
79. C ZONE CORRECTION WILL NOT BE MADE AND HONK AND CLS WILL
80. C NOT BE CALCULATED.
81. C * NOTE: LOCAL + KTZ = GMT; I.E. VALUES WEST OF GREENWICH ARE PLUS.
82. C DESC = DESCRIPTION OF GRAVITY STATION SITE IN ALPHA-NUM FORMAT
83. C
84. C DRFTCB = THE CORRECTION FACTOR FOR DRIFT OF THE GRAVITY METER.
85. C IF DRFTCB = 0.0 WE ASSUME A DRIFT RATE OF .003 MGALS/DAY
86. C IF ISW(5) EQUALS 1 PROGRAM DOES NOT MAKE DRIFT CORRECTION
87. C ASDFT = ASSUMED DRIFT FOR GRAVITY METER
88. C * NOTE: IF THE GRAVITY METER DRIFT IS NEGATIVE,
89. C THE CORRECTION FOR DRIFT IS A POSITIVE NUMBER.
90. C BASEG = TOTAL FIELD GRAVITY VALUE AT STATION OF REFERENCE
91. C THE VALUE IS READ IN WITH A FORMAT OF F3.0,F6.2 FROM
92. C WHICH BASE FOR OUTPUT AND BASG7 FOR COMPUTATION IS FORMED.
93. C GBSG7 HAS 977000. SUBTRACTED FROM IT ... FOR OUTPUT, CONVERT
94. C TO 13.F6.2 AFTER ADDING 977000.
95. C LSRCD = DEFAULT SOURCE CODE FOR GSM OUTPUT DATA
96. C DLAT, DLON = LATITUDE AND LONGITUDE IN DECIMAL DEGREES
97. C * NOTE: LOCATIONS NORTH AND EAST ARE CONSIDERED AS POSITIVE;
98. C SOUTH AND WEST ARE CONSIDERED NEGATIVE. (THIS IS THE
99. C EXACT OPPOSITE OF THE TIME ZONE CONVENTION)
100. C JTAPE = UNIT REF. NO. FOR PUNCHED CARD OUTPUT (SSW(2) OPTION)
101. C KTAPE = UNIT REF. NO. FOR GSM FORMAT OUTPUT (SSW(4) OPTION)
102. C IPCS = PUNCH CARD SKIP (THIS IS DONE FOR THE GRAVITY DESCRIPTION
103. C PUNCHED CARD)
104. C VALM = GRAVITY METER CALIBRATION TABLES ARRAY
105. C THIS IS THE TABLE USED TO 'LOOK-UP' OR CONVERT A GRAVITY METER
106. C COUNTER READING TO AN EQUIVALENT RELATIVE MILLIGAL VALUE.
107. C THE COUNTER READING IS READ WITH A FORMAT OF F8.3, FOR
108. C INSTANCE 3572.256
109. C THE HIGH-ORDER TWO DIGITS -- IN THIS CASE '35' -- ARE USED
110. C AS THE ARRAY INDEX. THE VALUE STORED IN VALM(35) IS THE
111. C EQUIVALENT MILLIGAL VALUE FOR A COUNTER READING OF 3500.000,
112. C SO WE INTERPOLATE A VALUE BETWEEN VALM(35) AND VALM(36) AND
113. C ARRIVE AT A RELATIVE MILLIGAL VALUE FOR 3572.256
114. C
115. C
116. C *****
117. C *
118. C * INITIALIZATION
119. C *

```

```

CR2G0530
CR2G0540
CR2G0550
CR2G0560
CR2G0570
CR2G0580
CR2G0590
CR2G0600
CR2G0610
CR2G0620
CR2G0630
CR2G0640
CR2G0650
CR2G0660
CR2G0670
CR2G0680
CR2G0690
CR2G0700
CR2G0710
CR2G0720
CR2G0730
CR2G0740
CR2G0750
CR2G0760
CR2G0770
CR2G0780
CR2G0790
CR2G0800
CR2G0810
CR2G0820
CR2G0830
CR2G0840
CR2G0850
CR2G0860
CR2G0870
CR2G0880
CR2G0890
CR2G0900
CR2G0910
CR2G0920
CR2G0930
CR2G0940
CR2G0950
CR2G0960
CR2G0970
CR2G0980
CR2G0990
CR2G1000
CR2G1010
CR2G1020
CR2G1030
CR2G1040
CR2G1050
CR2G1060
CR2G1070
CR2G1080
CR2G1090
CR2G1100
CR2G1110
CR2G1120

```

120.	C *****	CR2G1130
121.	C	CR2G1140
122.	DIMENSION KDATE(4)	CR2G1150
123.	DIMENSION VALM(70),BASEG(2),DESC(31),IGM(2)	CR2G1160
124.	DOUBLE PRECISION DTD	CR2G1170
125.	DOUBLE PRECISION DEC,RLAT,RAD,RLONG	CR2G1180
126.	DATA WEST,SOUTH/IW ','S '/	CR2G1190
127.	C	CR2G1200
128.	C * ASSUMED VALUES FOR L&R G=18 METER	CR2G1210
129.	C	CR2G1220
130.	DATA IBLNK,ILR,IG18/' ','L&R '','G=18'/	CR2G1230
131.	LSRCD = 006	CR2G1240
132.	ASDFT = .003	CR2G1250
133.	C	CR2G1260
134.	C * GET DATE OF RUN	CR2G1270
135.	C	CR2G1280
136.	CALL TODAY (KDATE)	CR2G1290
137.	C	CR2G1300
138.	IIN=105	CR2G1310
139.	IOUT=108	CR2G1320
140.	JTAPE=106	CR2G1330
141.	KTAPE = 2	CR2G1340
142.	ICBUNT=0	CR2G1350
143.	IPAGE = 0	CR2G1360
144.	C	CR2G1370
145.	IREC=1	CR2G1380
146.	IDEP = 0	CR2G1390
147.	RFA = 0.0	CR2G1400
148.	IREGC = 0	CR2G1410
149.	TCORR=99.9	CR2G1420
150.	LELC=09	
151.	LGC=01	
152.	IFFC=3	CR2G1450
153.	IFBC=0	CR2G1460
154.	C	CR2G1470
155.	C * READ IN GRAVITY METER CALIBRATION TABLE	CR2G1480
156.	C	CR2G1490
157.	DO 210 K = 1,70	CR2G1500
158.	READ (IIN,5150) J,TABLE	CR2G1510
159.	VALM(J)=TABLE	CR2G1520
160.	210 CONTINUE	CR2G1530
161.	C	CR2G1540
162.	C * INITIALIZE SENSE SWITCHES	CR2G1550
163.	C	CR2G1560
164.	INN = (ISW(=2))	CR2G1570
165.	C	CR2G1580
166.	C * READ GRAVITY METER TYPE, DRIFT VALUE, AND SOURCE CODE VALUE	CR2G1590
167.	C * SET UP DRIFT CORRECTION VALUE	CR2G1600
168.	C IF VALUE FEAD FROM CARD IS 0, USE ASDFT UNLESS SSW(5) IS 0N.	CR2G1610
169.	C	CR2G1620
170.	READ (IIN,5230) IGM(1),IGM(2),DRFTC0,LSRC,IELC,IGC	CR2G1630
171.	IF(DRFTC0.EQ.0.0) DRFTC0=ASDFT	CR2G1640
172.	IF(ISW(5).EQ.1) DRFTC0=0.0	CR2G1650
173.	IF (IGM(1).EQ.IBLNK.AND.IGM(2).EQ.IBLNK)	CR2G1660
174.	1 IGM(1) = ILR, IGM(2) = IG18	CR2G1670
175.	IF (LSRC.EQ. 0) LSRC = LSRCD	CR2G1680
176.	IF(IELC.EQ.0) IELC=LELC	
177.	IF(IGC.EQ.0) IGC=LGC	
178.	C	CR2G1690
179.	C * WRITE OUT JOB INITIALIZATION VALUES	CR2G1700

180.	C		CR2G1710
181.		WRITE (IOUT,5200) IPAGE,KDATE	CR2G1720
182.		WRITE (IOUT,5285) IGM(1),IGM(2),DRFTC0,LSRC,IELC,IGC	CR2G1730
183.		DB 294 II = 1,70	CR2G1740
184.		WRITE (IOUT,5290) II,VALM(II)	CR2G1750
185.		294 CONTINUE	CR2G1760
186.	C		CR2G1770
187.	C	*****	CR2G1780
188.	C	*	CR2G1790
189.	C	* COMPUTATION OF OBSERVED GRAVITY AND ANOMALIES	CR2G1800
190.	C	*	CR2G1810
191.	C	* BEGIN A GROUP OF MEASUREMENTS	CR2G1820
192.	C	*	CR2G1830
193.	C	*****	CR2G1840
194.	C		CR2G1850
195.	C	* READ BASE GRAVITY	CR2G1860
196.	C	AND CONVERT INTO UNITS COMPATIBLE WITH BOTH SYSTEMS	CR2G1870
197.	C		CR2G1880
198.		300 CONTINUE	CR2G1890
199.		READ (IIN,5320, END=910) BASEG(1),BASEG(2)	CR2G1900
200.		READ (IIN,5330, END=910) DENSE	CR2G1910
201.		BASG7=((BASEG(1)-977.)*1000.)*BASEG(2)	CR2G1920
202.		IBASE=BASEG(1)	CR2G1930
203.	C		CR2G1940
204.	C	* READ COUNTER READING CARDS FOR INDIVIDUAL STATIONS	CR2G1950
205.	C	THE FIRST CARD READ IS THE ONE FOR THE REFERENCE STATION	CR2G1960
206.	C	DRIFT IS COMPUTED FROM DATE ON THIS FIRST CARD.	CR2G1970
207.	C	ALL OTHER READINGS ARE REFERENCED TO THIS MEASUREMENT.	CR2G1980
208.	C		CR2G1990
209.		DB 890 I = 1,9000	CR2G2000
210.		READ (IIN,5405, END=910)	CR2G2010
211.		1 LSTAT,LDAY,M0,LYR,LTIME,CRN,LAT,RLATM,S0RN,L0NG,	CR2G2020
212.		IRL0M,W0RE,ELEV,KTZ ,(DESC(IK),IK=1,31)	CR2G2030
213.		IPCS=0	CR2G2040
214.		CLS=.99	CR2G2050
215.		H0NK=.99	CR2G2060
216.		CR=CRN	CR2G2070
217.		NCR=CRN	CR2G2080
218.	C		CR2G2090
219.	C	* CHECK TO SEE IF HAVE NEW REFERENCE STATION OR CALL EXIT	CR2G2100
220.	C	(STMT #180 IS EXIT; #300 IS START OF NEW GROUP)	CR2G2110
221.	C		CR2G2120
222.		IF(NCR)500,500,420	CR2G2130
223.		500 IF(LYR)415,180,415	CR2G2140
224.		415 I=1	CR2G2150
225.		G0T0 300	CR2G2160
226.	C		CR2G2170
227.	C	* SEARCH TABLES FOR GRAVITY VALUE CORRESPONDING TO COUNTER READING	CR2G2180
228.	C		CR2G2190
229.		420 TUCR=CR*0.01	CR2G2200
230.		LC=TUCR	CR2G2210
231.		CI=LC	CR2G2220
232.		CI=CI*100.	CR2G2230
233.		C2=CR-CI	CR2G2240
234.		RELM=VALM(LC)+(C2*.01*(VALM(LC+1)-VALM(LC)))	CR2G2250
235.	C		CR2G2260
236.	C	* CALCULATE LAT AND L0N IN RADIAN AND IN DECIMAL DEGREES	CR2G2270
237.	C		CR2G2280
238.		RDEG=LAT	CR2G2290
239.		DEC=RLATM*1.666666E -2	CR2G2300

240.	RLAT=RDEG+DEC	CR2G2310
241.	DLAT = RLAT	CR2G2320
242.	RAD=RLAT*(1.7453293D-2)	CR2G2330
243.	RLAT=RAD	CR2G2340
244.	C	CP2G2350
245.	RDEG=RLONG	CR2G2360
246.	DEC=RLONG*1.666666666D-2	CR2G2370
247.	RLONG=RDEG + DEC	CR2G2380
248.	DLON = RLONG	CR2G2390
249.	RLONG=RLONG*1.7453293 D-2	CR2G2400
250.	C	CR2G2410
251.	C * IF KTZ IS EQUAL TO 99 MEANS HAVE NOT MADE OR LOOKED UP THE	CR2G2420
252.	C TIME ZONE CORRECTION. THERE THE FOLLOWING CALCULATIONS	CR2G2430
253.	C ARE NOT NEEDED BECAUSE WE CANNOT CALCULATE THE TIDAL OR	CR2G2440
254.	C HONKLE CORRECTIONS WITHOUT IT	CR2G2450
255.	C	CR2G2460
256.	IF(KTZ.NE.99) GO TO 610	CR2G2470
257.	KGDA=LDAY	CR2G2480
258.	KGM0=M0	CR2G2490
259.	KGYR=LYR	CR2G2500
260.	KGHM=LTIME	CR2G2510
261.	ID=0	CR2G2520
262.	GO TO 630	CR2G2530
263.	C	CR2G2540
264.	610 CONTINUE	CR2G2550
265.	KTZ=KTZ	CR2G2560
266.	CALL CHGMT(LDAY,M0,LYR,LTIME,KTZ,KGDA,KGM0,KGYR,KGHM,NTZ)	CR2G2570
267.	CALL M2DY(KGYR,KGM0,KGDA,ID)	CR2G2580
268.	ADAY = KGHM	CR2G2590
269.	ADAY = ADAY / 2400.	CR2G2600
270.	ADAY = ADAY + FLOAT (ID)	CR2G2610
271.	C	CR2G2620
272.	C NORTH LAT OR EAST LON IS POSITIVE	CR2G2630
273.	C SOUTH LAT OR WEST LON IS NEGATIVE	CR2G2640
274.	C	CR2G2650
275.	IF (RLAT) 620,619,620	CR2G2660
276.	619 IF (RLONG) 620,630,620	CR2G2670
277.	620 CONTINUE	CR2G2680
278.	C	CR2G2690
279.	IF(SORN.EQ.SOUTH) RLAT=-RLAT; DLAT=-DLAT	CR2G2700
280.	IF(WORE.EQ.WEST) RLONG=-RLONG; DLON=-DLON	CR2G2710
281.	C	CR2G2720
282.	C * CALCULATE CLS AND HONK VALUES	CR2G2730
283.	C	CR2G2740
284.	IHR=KGHM/100	CR2G2750
285.	IMIN=KGHM-IHR*100	CR2G2760
286.	CALL TIDAL(RLAT,RLONG,KGYR,ID,IHR,IMIN,CLS,HONK,DTD)	CR2G2770
287.	RELM=RELM+CLS+HONK	CR2G2780
288.	C	CR2G2790
289.	630 CONTINUE	CR2G2800
290.	IF(I=1,440,440,450	CR2G2810
291.	C	CR2G2820
292.	C * NEW REFERENCE STATION (= FIRST CARD OF GROUP) PROCESSING	CR2G2830
293.	C	CR2G2840
294.	440 REF=RELM	CR2G2850
295.	IDA1=KGDA	CR2G2860
296.	IM01=KGM0	CR2G2870
297.	IYR1=KGYR	CR2G2880
298.	ITH1=KGHM	CR2G2890
299.	LSTA=LSTAT	CR2G2900

```

300.      IPCS=1                                CR2G2910
301.      LCNT = 0                             CR2G2920
302.      IPAGE = 1                           CR2G2930
303.      C                                     CR2G2940
304.      C * CALCULATE DRIFT                 CR2G2950
305.      C                                     CR2G2960
306.      450 DIFFR=RELM=REF                   CR2G2970
307.      CALL CDATE(IDA1,IM01,IYR1,ITM1,KGDA,KGM0,KGYR,KGHM,TIMD) CR2G2980
308.      DRIFT=(TIMD/24.0)*DRFTC0             CR2G2990
309.      G0BS7=BASG7+DIFFR-DRIFT             CR2G3000
310.      IC0UNT=IC0UNT+1                     CR2G3010
311.      C                                     CR2G3020
312.      C * PUT G0BS7 + 977000. INTO OUTPUT UNITS CR2G3030
313.      C                                     CR2G3040
314.      IC0N=G0BS7/1000.                   CR2G3050
315.      TEMP=IC0N*1000                     CR2G3060
316.      G0BS=G0BS7-TEMP                    CR2G3070
317.      IG=IC0N+977                        CR2G3080
318.      C                                     CR2G3090
319.      C * COMPUTE FREE-AIR AND BOUGUER ANOMALIES CR2G3100
320.      C                                     CR2G3110
321.      X=2.*RAD                           CR2G3120
322.      C2R=C0S(X)                         CR2G3130
323.      FELEV=((0.30855+0.00022*C2R)*ELEV)*(((ELEV*0.001)**2)*0.072) CR2G3140
324.      RA1=RAD                             CR2G3150
325.      GFREE=G0BS7-GINTF(RA1)+FELEV
326.      BELEV=0.04185*DENSE*ELEV
327.      G0BUG=GFREE-BELEV                  CR2G3170
328.      C                                     CR2G3180
329.      C * CHECK TO SEE IF LAT AND L0N = 0 CR2G3190
330.      C 0R IF ELEV = 0                   CR2G3200
331.      C IF THEY D0 SET GFREE AND G0BUG EQUAL TO 999.0 CR2G3210
332.      C                                     CR2G3220
333.      IF(LAT)2050,2049,2050              CR2G3230
334.      2049 IF(RLATM) 2050,2051,2050      CR2G3240
335.      2051 IF(L0NG) 2050,2052,2050      CR2G3250
336.      2052 IF(RL0M) 2050,2053,2050      CR2G3260
337.      C                                     CR2G3270
338.      2050 IF (ELEV) 2054,2053,2054     CR2G3280
339.      2053 GFREE=999.0                    CR2G3290
340.      G0BUG=999.0                        CR2G3300
341.      2054 CONTINUE                      CR2G3310
342.      C                                     CR2G3320
343.      NSTATN=LSTAT                       CR2G3330
344.      C                                     CR2G3340
345.      C                                     CR2G3350
346.      C * COMPUTE GSUM SORT KEY FIELDS   CR2G3360
347.      C                                     CR2G3370
348.      PLAT = DLAT + 90.                  CR2G3380
349.      LTKEY = PLAT                       CR2G3390
350.      PL0N = DL0N + 180.                 CR2G3400
351.      LGKEY = PL0N                       CR2G3410
352.      IAKEY = 0                          CR2G3420
353.      C                                     CR2G3430
354.      C *****                          CR2G3440
355.      C *                                  CR2G3450
356.      C * OUTPUT THE DESIRED INFORMATION CR2G3460
357.      C *                                  CR2G3470
358.      C *****                          CR2G3480
359.      C                                     CR2G3490

```

360.	C * PUNCH OUTPUT FOR INPUT TO STATION DESCRIPTION PROGRAM	CR2G3510
361.	C IPCS PREVENTS US FROM PUNCHING THE REFERENCE STATION WHEN	CR2G3520
362.	C IT IS THE FIRST CARD OUT	CR2G3530
363.	C	CR2G3540
364.	IF (IPCS.EQ.1) GOTO 835	CR2G3550
365.	IF (ISW(2)) 835,825,835	CR2G3560
366.	825 WRITE (JTAPE,5825) LYR,M0,LDAY,LTIME,LSTAT,IGM(1),IGM(2),	CR2G3570
367.	1 IG,G0BS,LSTA,IBASE,BASEG(2),RLAT,S0RN,RLONG,W0RE,ELEV	CR2G3580
368.	835 CONTINUE	CR2G3590
369.	C	CR2G3600
370.	C * LISTING OF CALCULATED VALUES	CR2G3610
371.	C	CR2G3620
372.	IF (ISW(1)) 869,851,869	CR2G3630
373.	851 IF (LCNT) 852,852,860	CR2G3640
374.	C	CR2G3650
375.	C PRINT PAGE HEADING	CR2G3660
376.	C	CR2G3670
377.	852 WRITE (IOUT,5001)	CR2G3680
378.	IF (IPAGE - 1) 853,853,854	CR2G3690
379.	853 WRITE (IOUT,5853)	CR2G3700
380.	854 CONTINUE	CR2G3710
381.	WRITE (IOUT,5200) IPAGE,KDATE	CR2G3720
382.	WRITE (IOUT,5855) LSTA, IDA1,IM01,IYR1,ITM1,IGM(1),IGM(2),LSRC	CR2G3730
383.	WRITE (IOUT,5856) IBASE,BASEG(2), REF, DENSE, DRFTC0	CR2G3740
384.	WRITE (IOUT,5858)	CR2G3750
385.	IPAGE = IPAGE + 1	CR2G3760
386.	LCNT = 39	CR2G3770
387.	C	CR2G3780
388.	860 WRITE (IOUT,5860)	CR2G3790
389.	1 LSTAT,LDAY,M0,LYR,LTIME,KTZ,LAT,RLATM,S0RN,CR,	CR2G3800
390.	2 GFREE,CLS ,DIFFR,DESC	CR2G3810
391.	WRITE (IOUT,5862)	CR2G3820
392.	1 IG,G0BS,KGDA,KGM0,KGYR,KGHM,ELEV,L0NG,RL0M,W0RE,RELM,	CR2G3830
393.	2 GB0UG,W0NK,DRIFT,TIMD,ADAY	CR2G3840
394.	LCNT = LCNT + 3	CR2G3850
395.	869 CONTINUE	CR2G3860
396.	C	CR2G3870
397.	C * OUTPUT AT GSUM FORMAT TO KTAPE	CR2G3880
398.	C FORMAT FORWARD CODE = 3	CR2G3890
399.	C	CR2G3900
400.	IF (ISW(4)) 889,871,889	CR2G3910
401.	871 WRITE (KTAPE,5871) IREC,LSRC,KGDA,KGM0,KGYR,KGHM,DLAT,DL0N,ELEV,	CR2G3920
402.	1 IG,G0BS,IDEF,GFREE,GB0UG,TC0RR,IELC,IGC,RFA,IREGC,	CR2G3930
403.	2 IFFC,CLS,H0NK,CRN,(DESC(II),II=1,16),NSTATN,IFBC,	CR2G3940
404.	3 LTKEY,LGKEY,IAKEY	CR2G3950
405.	889 CONTINUE	CR2G3960
406.	C	CR2G3970
407.	C * STMT. #890 IS THE END OF THE READ DO=LOOP	CR2G3980
408.	C	CR2G3990
409.	890 CONTINUE	CR2G4000
410.	C	CR2G4010
411.	C *****	CR2G4020
412.	C *	CR2G4030
413.	C * END OF JOB	CR2G4040
414.	C *	CR2G4050
415.	C *****	CR2G4060
416.	C	CR2G4070
417.	180 CONTINUE	CR2G4080
418.	181 CONTINUE	CR2G4090
419.	910 CONTINUE	CR2G4100



NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
ADAY	R	SCALR	000BE V	1	ASFT	R	SCALR	0007E V	1	BASG	R	SCALR	0007E V	1
BASG7	R	SCALR	00098 V	1	BELEV	R	SCALR	000D5 V	1	CDATE	R	SCALR	000B8 V	1
CHGMT	R	SPR8G	00000 V	1	CI	R	SCALR	000B1 V	1	CLS	R	SCALR	000A0 V	1
CBS	R	SPR8G	00000 V	1	CR	R	SCALR	000AD V	1	CLN	R	SCALR	0007D V	2
C2	R	SCALR	00082 V	1	C2R	R	SCALR	000D1 V	1	DEC	D	SCALR	000C8 V	1
DENSE	R	SCALR	00097 V	1	DESC	R	SCALR	0004C V	31	DIFFR	R	SCALR	00092 V	1
DLAT	R	SCALR	000B5 V	1	DLN	R	SCALR	000B6 V	1	DRFTC9	R	SCALR	000A7 V	1
DRIFT	R	SCALR	000CA V	1	DTD	D	SCALR	0006E V	2	ELEV	R	SCALR	000C6 V	1
FELEV	R	SCALR	000D2 V	1	FLAT	R	SPR8G	00000 V	1	GBUG	R	SCALR	000C6 V	1
GFREE	R	SCALR	000D4 V	1	GINTF	R	SPR8G	00000 V	1	GBUS	R	SCALR	000C6 V	1
GBBS7	R	SCALR	000CB V	1	HONK	R	SCALR	000AC V	1	I	I	SCALR	0009A V	1
IAKEY	R	SCALR	000DC V	1	IBASE	R	SCALR	00099 V	1	IBLANK	I	SCALR	0007A V	1
ICBN	I	SCALR	000CC V	1	ICOUNT	I	SCALR	00083 V	1	ID	I	SCALR	00088 V	1
IDAI	I	SCALR	000C2 V	1	IDEP	I	SCALR	00086 V	1	IELC	I	SCALR	00094 V	1
IFBC	I	SCALR	0008D V	1	IFFC	I	SCALR	0008C V	1	IG	I	SCALR	000CF V	1
IGC	I	SCALR	00095 V	1	IGH	I	SCALR	0008B V	2	IG18	I	SCALR	0007C V	1
IHR	I	SCALR	000BF V	1	I1	I	SCALR	00096 V	1	IIN	I	SCALR	0007F V	1
IK	I	SCALR	000A9 V	1	IIR	I	SCALR	0007B V	1	IMIN	I	SCALR	000C0 V	1
IM01	I	SCALR	000C3 V	1	INN	I	SCALR	00091 V	1	IOUT	I	SCALR	00080 V	1
IPAGE	I	SCALR	00084 V	1	IPCS	I	SCALR	000AA V	1	IREC	I	SCALR	00085 V	1
IREGC	I	SCALR	00088 V	1	ISN	I	SPR8G	00000 V	1	ITM1	I	SCALR	000C5 V	1
IYR1	I	SCALR	000C4 V	1	JDATE	I	SCALR	0008F V	1	JTAPE	I	SCALR	00081 V	1
K	I	SCALR	0008E V	1	KDATE	I	SCALR	00000 V	4	KGDA	I	SCALR	00087 V	1
KGMH	I	SCALR	0008A V	1	KGM0	I	SCALR	00088 V	1	KGYR	I	SCALR	00089 V	1
KTAPE	I	SCALR	00082 V	1	KTY	I	SCALR	0008C V	1	KTZ	I	SCALR	00088 V	1
LAT	I	SCALR	000A1 V	1	LC	I	SCALR	00080 V	1	LCNT	I	SCALR	000C7 V	1
LDAY	I	SCALR	0009C V	1	LELC	I	SCALR	0008A V	1	LGC	I	SCALR	000C7 V	1
LGKEY	I	SCALR	000DB V	1	LNG	I	SCALR	00044 V	1	LSRC	I	SCALR	00093 V	1
LSRCD	I	SCALR	0007D V	1	LSTA	I	SCALR	000C6 V	1	LSTAT	I	SCALR	0009B V	1
LTIME	I	SCALR	0009F V	1	LTKEY	I	SCALR	000D9 V	1	LYR	I	SCALR	0009E V	1
M8	I	SCALR	0009D V	1	M2DY	I	SPR8G	00000 V	1	NCR	I	SCALR	000AE V	1
NSTATN	I	SCALR	000D7 V	1	NIZ	I	SCALR	0008D V	1	PLAT	R	SCALR	000D8 V	1
PLBN	I	SCALR	000DA V	1	NIZ	I	SCALR	00074 V	2	RAI	R	SCALR	000D3 V	1
RDEG	R	SCALR	00084 V	1	REF	R	SCALR	000C1 V	1	RELH	R	SCALR	00083 V	1
RFA	R	SCALR	00087 V	1	RLAT	R	SCALR	00072 V	1	RLM	R	SCALR	000A2 V	1
RLGH	R	SCALR	000A5 V	1	RLNG	R	SCALR	00076 V	2	SRN	R	SCALR	000A3 V	1
SETH	R	SCALR	00079 V	1	TABLE	R	SCALR	00090 V	1	TGRR	R	SCALR	00089 V	1
TEMP	R	SCALR	000CD V	1	TIDAL	R	SPR8G	00000 V	1	TIND	R	SCALR	000C9 V	1
T9DAY	R	SPR8G	00000 V	1	TUCR	R	SCALR	000AF V	1	VALH	R	SCALR	000C9 V	1
WEST	R	SCALR	00078 V	1	WRE	R	SCALR	000A6 V	1	X	R	SCALR	000D0 V	70

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
24	002B9	180	0029F	210	0003F	294	0008F	300	00093
415	000C0	420	000DF	450	00188	500	000DA	595	00283
610	00127	619	00148	630	00173	825	001F6	835	00208
851	00210	852	00212	854	0021D	860	00241	869	0026A
871	0026F	889	0029B	910	0029F	2049	001D2	2050	001D8
2051	001B4	2052	001D6	2054	001DE	5001	002C5	5150	002C8
5200	002CC	5230	002D6	5290	002F6	5320	002FE	5330	00302
5405	00305	5825	00313	5855	00322	5856	0034D	5858	00365
5860	00339	5862	003AE	5950	003E0				

## LOCAL VARIABLES (221 WORDS):

00000	KDATE	00004	VALM	00044	BASEG	0004C	DESC	0006B	IGM	0006E	DTD
00070	DEC	00072	RLAT	00074	RAD	00076	RLNG	00078	WEST	00079	SOUTH
0007A	IBLNK	0007B	ILR	0007C	IG18	0007D	LSRCD	0007E	ASDFT	0007F	ITN
00080	IBUT	00081	JTAPE	00082	KTAPE	00083	ICBUNT	00084	IPAGE	00085	IREC
00086	IDEP	00087	RFA	00088	IREGC	00089	TCORR	0008A	LELC	0008B	LGC
0008C	IFFC	0008D	IFBC	0008E	K	0008F	J	00090	TABLE	00091	INN
00092	DRFTCB	00093	LSRC	00094	IELC	00095	IGC	00096	II	00097	DENSE
00098	BASG7	00099	IBASE	0009A	I	0009B	LSTAT	0009C	LDAY	0009D	M8
0009E	LYR	0009F	LTIME	000AC	CRN	000A1	LAT	000A2	RLATM	000A3	SERN
000A4	L8NG	000A5	RL8M	000A6	W8RE	000A7	ELEV	000A8	KTZ	000A9	IK
000AA	IPCS	000AB	CLS	000AC	H8NK	000AD	CR	000AE	NCR	000AF	TUCR
000B0	LC	000B1	CI	000B2	C2	000B3	RELM	000B4	RDEG	000B5	DLAT
000B6	DL8N	000B7	KGDA	000B8	KGM8	000B9	KGYR	000BA	KGHM	000BB	ID
000BC	KTT	000BD	NTZ	000BE	ADAY	000BF	IHR	000C0	IMIN	000C1	REF
000C2	IDA1	000C3	IM81	000C4	IYR1	000C5	ITM1	000C6	LSTA	000C7	LCNT
000C8	DIFFR	000C9	TMD	000CA	DRIFT	000CB	G8BS7	000CC	IC8N	000CD	TEMP
000CE	G8BS	000CF	IG	000D0	X	000D1	C2R	000D2	FELEV	000D3	RA1
000D4	GFREE	000D5	BELEV	000D6	G88UG	000D7	NSTATN	000D8	PLAT	000D9	LTKEY
000DA	PL8N	000DB	LGKEY	000DC	IAKEY						

## BLANK COMMON (0 WORDS)

## INTRINSIC SUBPROGRAMS USED:

C8S      FLOAT

## EXTERNAL SUBPROGRAMS REQUIRED:

CDATE	CHGMT	GINTF	ISM	M2DY	TIDAL	T8DAY	F:101
F:102	F:103	F:104	F:105	F:106	F:108	M:08	M:0C
9BCDRDEE	9BCDREAD	9BCDWRIT	9C8S	9DT8R	9ENDFILE	9ENDI8L	9INITIAL
918DATA	918LUSA	91T8R	9PRINT	9REWIND	9RT8I	9STOP	

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	1001	DEC	HEX
CONSTANTS:	31	WORDS	----
LOCAL VARIABLES:	221		
TEMPS:	1		
			----
TOTAL PROGRAM:	1254		004E6

```

1.  C   PROGRAM CR2G67
2.  C   ORIGINAL VERSION 2 OCT 75
3.  C   MOD OF CR2G TO CALCULATE IGSN 67 VALUES
4.  C   VERSION OF 20 MAR 1973, TO ADD INPUT OF ELEV AND G METER CODES
5.  C   MODIFICATION BY C. BOWIN
6.  C   OUTPUT ' *** PGM CR2G OF 20 MAR 1973'
7.  C
8.  C * MAY 72 -- BY S. ABBOT TO CORRECT OUTPUT LISTING, AND TO
9.  C   OUTPUT GSUM DATA WITH LAT AND LON IN DECIMAL DEGREES
10. C   ALSO TO CLEAN UP THE COMMENTS AND SO ON
11. C * MOD 18 JAN 71/ -S.ABBOT TO CORRECT
12. C***** REWRITTEN IN A HOPEFUL MANNER BY J. WOLFE JUNE E 1969
13. C
14. C * INPUT
15. C
16. C * JOB INITIALIZATION
17. C
18. C   1) GRAVITY METER TABLES -- 70 CARDS
19. C   2) SENSE SWITCH CARD -- 8011 (SSW(0) = CC 80)
20. C   3) IGM(1), IGM(2), DRFTC0, LSRC (2A4,2X,F10.5,I5)
21. C   IGM -- NUMBER AND/OR MODEL OF GRAVITY METER USED
22. C   FOR INSTANCE -- L&R G-18
23. C   DRFTC0 -- DRIFT CORRECTION FOR GRAVITY METER -- F10.5
24. C   LSRC -- SOURCE CODE FOR GSUM OUTPUT DATA
25. C * NOTE: IF DRFTC0 = 0.0, THE ASSUMED VALUE FOR THE GRAVITY METER
26. C   DRIFT (ASDFT) WILL BE USED, UNLESS SSW(5) IS ON
27. C * NOTE: IF IGM(1) AND IGM(2) ARE BOTH BLANK, THE GRAVITY METER TYPE
28. C   WILL BE SET TO THE DEFAULT TYPE OF 'L&R G-18'
29. C * NOTE: IF LSRC = 0, THE SOURCE CODE WILL BE SET TO THE DEFAULT
30. C   VALUE -- LSRCD
31. C
32. C * THESE ARE FOLLOWED BY GROUPS OF INDIVIDUAL STATION COUNTER
33. C   READING CARDS. EACH GROUP IS HEADED BY 3 CARDS:
34. C   1) THE ABSOLUTE GRAVITY VALUE FOR THE REFERENCE STATION --
35. C   F3.0,F6.2 -- BASEG(1), BASEG(2)
36. C   2) THE CRUSTAL DENSITY IN GM PER CU CM TO BE USED IN
37. C   THE CALCULATION OF THE BOUGUER ANOMALY -- F4.2 -- DENSE
38. C   3) THE COUNTER READING CARD FOR THE REFERENCE STATION
39. C
40. C   FORMAT FOR COUNTER READING DATA IS THAT OF 17 MAY 1966
41. C
42. C * THESE ARE FOLLOWED BY COUNTER READING CARDS FOR THE REST OF THE
43. C   MEASUREMENTS THAT ARE TO BE TIED TO THE REFERENCE MEASUREMENT.
44. C
45. C * A COUNTER READING CARD WITH ALL ZEROS EXCEPT FOR THE
46. C   YEAR VALUE (CC 9,10) WILL CAUSE THE PGM TO BRANCH TO READ
47. C   NEW CARDS FOR BASEG AND DENSE AND THE REF STATION
48. C   A CARD WITH ALL ZEROS INCLUDING YEAR WILL GO TO E.8.J
49. C   DO NOT HAVE AN ALL-ZEROS CARD FOLLOWING A CARD WITH ONLY THE YEAR.
50. C
51. C * SENSE SWITCH OPTIONS
52. C
53. C   SSW(1) OFF, FOR PRINTED OUTPUT OF COMPUTED VALUES FOR EACH STATION
54. C   ON, FOR SUPPRESSION OF PRINTED OUTPUT
55. C   SSW(2) OFF, TO PUNCH OUTPUT FOR INPUT TO GRAVITY DESCR. PGM. 'GDS'
56. C   ON, TO SUPPRESS PUNCHED OUTPUT
57. C   SSW(4) OFF, TO OUTPUT GSUM FORMATTED DATA TO 'KTAPE'
58. C   ON, TO SUPPRESS GSUM FORMAT OUTPUT

```

```

CR2G0010
CR2G0020
CR2G0030
CR2G0040
CR2G0050
CR2G0060
CR2G0070
CR2G0080
CR2G0090
CR2G0100
CR2G0110
CR2G0120
CR2G0130
CR2G0140
CR2G0150
CR2G0160
CR2G0170
CR2G0180
CR2G0190
CR2G0200
CR2G0210
CR2G0220
CR2G0230
CR2G0240
CR2G0250
CR2G0260
CR2G0270
CR2G0280
CR2G0290
CR2G0300
CR2G0310
CR2G0320
CR2G0330
CR2G0340
CR2G0350
CR2G0360
CR2G0370
CR2G0380
CR2G0390
CR2G0400
CR2G0410
CR2G0420
CR2G0430
CR2G0440
CR2G0450
CR2G0460
CR2G0470
CR2G0480
CR2G0490
CR2G0500

```

60.	C	SSW(5) OFF, FOR GRAVITY METER DRIFT CORRECTION	CR2G0510
61.	C	ON, FOR SUPPRESSION OF DRIFT CORRECTION	CR2G0520
62.	C		CR2G0530
63.	C		CR2G0540
64.	C	* VARIABLE DEFINITIONS	CR2G0550
65.	C		CR2G0560
66.	C	LDAY, MB, LYR, LTIME = DATE AND TIME OF READING -- IF LOCAL TIME	CR2G0570
67.	C	IS USED, KTZ SHOULD ALSO BE ENTERED. IF GMT, KTZ ALWAYS = 0	CR2G0580
68.	C	THIS IS THE TIME AS READ FROM THE COUNTER READING RECORDS.	CR2G0590
69.	C	KGDA, KGM0, KGYR, KGWM = DATE AND TIME OF READING -- IN GMT (OR	CR2G0600
70.	C	LOCAL IF KTZ = 99)	CR2G0610
71.	C	IDA1, IM01, IYR1, ITM1 = DATE AND TIME OF REFERENCE STATION	CR2G0620
72.	C	READING (IN GMT (OR LOCAL IF KTZ = 99))	CR2G0630
73.	C	CRN = GRAVITY METER COUNTER READING (IN UNITS)	CR2G0640
74.	C	ELEV = ELEVATION OF GRAVITY METER ( IN METERS)	CR2G0650
75.	C	KTZ = THE TIME ZONE CORRECTION. IF KTZ = 99, IT INDICATES	CR2G0660
76.	C	THAT KTZ WAS NOT AVAILABLE. THUS CLS AND H0NK = 0.0	CR2G0670
77.	C	DATE/TIME MAY BE ENTERED AS GMT, IN WHICH CASE KTZ	CR2G0680
78.	C	WILL ALWAYS BE EQUAL TO ZERO.	CR2G0690
79.	C	IF TIME OF READING IS IN LOCAL TIME BUT TIME ZONE IS NOT	CR2G0700
80.	C	KNOWN, KTZ MAY BE ENTERED AS 99, IN WHICH CASE A TIME	CR2G0710
81.	C	ZONE CORRECTION WILL NOT BE MADE AND H0NK AND CLS WILL	CR2G0720
82.	C	NOT BE CALCULATED.	CR2G0730
83.	C	* NOTE: LOCAL + KTZ = GMT; I.E. VALUES WEST OF GREENWICH ARE PLUS.	CR2G0740
84.	C	DESC = DESCRIPTION OF GRAVITY STATION SITE IN ALPHA-NUM FORMAT	CR2G0750
85.	C		CR2G0760
86.	C	DRFTC0 = THE CORRECTION FACTOR FOR DRIFT OF THE GRAVITY METER.	CR2G0770
87.	C	IF DRFTC0 = 0.0 WE ASSUME A DRIFT RATE OF .003 MGALS/DAY	CR2G0780
88.	C	IF ISW(5) EQUALS 1 PROGRAM DOES NOT MAKE DRIFT CORRECTION	CR2G0790
89.	C	ASDFT = ASSUMED DRIFT FOR GRAVITY METER	CR2G0800
90.	C	* NOTE: IF THE GRAVITY METER DRIFT IS NEGATIVE,	CR2G0810
91.	C	THE CORRECTION FOR DRIFT IS A POSITIVE NUMBER.	CR2G0820
92.	C	BASEG = TOTAL FIELD GRAVITY VALUE AT STATION OF REFERENCE	CR2G0830
93.	C	THE VALUE IS READ IN WITH A FORMAT OF F3.0,F6.2 FROM	CR2G0840
94.	C	WHICH IBASE FOR OUTPUT AND BASG7 FOR COMPUTATION IS FORMED.	CR2G0850
95.	C	G0BS7 HAS 977000. SUBTRACTED FROM IT ... FOR OUTPUT, CONVERT	CR2G0860
96.	C	TO I3,F6.2 AFTER ADDING 977000.	CR2G0870
97.	C	LSRCD = DEFAULT SOURCE CODE FOR GSM OUTPUT DATA	CR2G0880
98.	C	DLAT, DL0N = LATITUDE AND LONGITUDE IN DECIMAL DEGREES	CR2G0890
99.	C	* NOTE: LOCATIONS NORTH AND EAST ARE CONSIDERED AS POSITIVE;	CR2G0900
100.	C	SOUTH AND WEST ARE CONSIDERED NEGATIVE. (THIS IS THE	CR2G0910
101.	C	EXACT OPPOSITE OF THE TIME ZONE CONVENTION)	CR2G0920
102.	C	JTAPE = UNIT REF. NO. FOR PUNCHED CARD OUTPUT (SSW(2) OPTION)	CR2G0930
103.	C	KTAPE = UNIT REF. NO. FOR GSM FORMAT OUTPUT (SSW(4) OPTION)	CR2G0940
104.	C	IPCS = PUNCH CARD SKIP (THIS IS DONE FOR THE GRAVITY DESCRIPTION	CR2G0950
105.	C	PUNCHED CARD)	CR2G0960
106.	C	VALM = GRAVITY METER CALIBRATION TABLES ARRAY	CR2G0970
107.	C	THIS IS THE TABLE USED TO LOOK-UP OR CONVERT A GRAVITY METER	CR2G0980
108.	C	COUNTER READING TO AN EQUIVALENT RELATIVE MILLIGAL VALUE.	CR2G0990
109.	C	THE COUNTER READING IS READ WITH A FORMAT OF F8.3 , FOR	CR2G1000
110.	C	INSTANCE 3572.256	CR2G1010
111.	C	THE HIGH-ORDER TWO DIGITS -- IN THIS CASE '35' -- ARE USED	CR2G1020
112.	C	AS THE ARRAY INDEX. THE VALUE STORED IN VALM(35) IS THE	CR2G1030
113.	C	EQUIVALENT MILLIGAL VALUE FOR A COUNTER READING OF 3500.000,	CR2G1040
114.	C	SO WE INTERPOLATE A VALUE BETWEEN VALM(35) AND VALM(36) AND	CR2G1050
115.	C	ARRIVE AT A RELATIVE MILLIGAL VALUE FOR 3572.256	CR2G1060
116.	C		CR2G1070
117.	C		CR2G1080
118.	C	*****	CR2G1090
119.	C	*	CR2G1100

120.	C * INITIALIZATION	CR2G1110
121.	C *	CR2G1120
122.	C *****	CR2G1130
123.	C	CR2G1140
124.	DIMENSION KDATE(4)	CR2G1150
125.	DIMENSION VALM(70),BASEG(2),DESC(31),IGM(2)	CR2G1160
126.	DOUBLE PRECISION DTD	CR2G1170
127.	DOUBLE PRECISION DEC,RLAT,RAD,RLONG	CR2G1180
128.	DATA WEST,SOUTH/IW '1',IS '1/'	CR2G1190
129.	C	CR2G1200
130.	C * ASSUMED VALUES FOR L&R G=18 METER	CR2G1210
131.	C	CR2G1220
132.	DATA IBLNK,ILR,IG18/' ','L&R ','IG=18'/'	CR2G1230
133.	OUTPUT 'PROGRAM CR2G67 VERSION 2 OCT 75'	
134.	LSRCD = 006	CR2G1240
135.	ASDFT = .003	CR2G1250
136.	C	CR2G1260
137.	C * GET DATE OF RUN	CR2G1270
138.	C	CR2G1280
139.	CALL T8DAY (KDATE)	CR2G1290
140.	C	CR2G1300
141.	IIN=105	CR2G1310
142.	IBUT=108	CR2G1320
143.	JTAPE=106	CR2G1330
144.	KTAPE = 2	CR2G1340
145.	ICOUNT=0	CR2G1350
146.	IPAGE = 0	CR2G1360
147.	C	CR2G1370
148.	IREC=1	CR2G1380
149.	IDEP = 0	CR2G1390
150.	RFA = 0.0	CR2G1400
151.	IREGC = 0	CR2G1410
152.	TCORR=99.9	CR2G1420
153.	LELC=09	
154.	LGC=01	
155.	IFFC=3	CR2G1450
156.	IFBC=0	CR2G1460
157.	C	CR2G1470
158.	C * READ IN GRAVITY METER CALIBRATION TABLE	CR2G1480
159.	C	CR2G1490
160.	DO 210 K = 1,70	CR2G1500
161.	READ (IIN,5150) J,TABLE	CR2G1510
162.	VALM(J)=TABLE	CR2G1520
163.	210 CONTINUE	CR2G1530
164.	C	CR2G1540
165.	C * INITIALIZE SENSE SWITCHES	CR2G1550
166.	C	CR2G1560
167.	INN = (ISW(-2))	CR2G1570
168.	C	CR2G1580
169.	C * READ GRAVITY METER TYPE, DRIFT VALUE, AND SOURCE CODE VALUE	CR2G1590
170.	C * SET UP DRIFT CORRECTION VALUE	CR2G1600
171.	C IF VALUE FEAD FROM CARD IS 0, USE ASDFT UNLESS SSW(5) IS ON.	CR2G1610
172.	C	CR2G1620
173.	READ (IIN,5230) IGM(1),IGM(2),DRFTCB,LSRC,IELC,IGC	CR2G1630
174.	IF (DRFTCB.EQ.0.0) DRFTCB=ASDFT	CR2G1640
175.	IF (ISW(5).EQ.1) DRFTCB=0.0	CR2G1650
176.	IF (IGM(1).EQ.IBLNK.AND.IGM(2).EQ.IBLNK)	CR2G1660
177.	2 IF IGM(1) = ILR; IGM(2) = IG18	CR2G1670
178.	IF (LSRC.EQ.0) LSRC = LSRCD	CR2G1680
179.	IF (IELC.EQ.0) IELC=LELC	

```

180.      IF(IGC,EO,0) IGC=LGC
181.      C
182.      C * WRITE OUT JOB INITIALIZATION VALUES
183.      C
184.          WRITE (IOUT,5200) IPAGE,KDATE
185.          WRITE (IOUT,5285) IGM(1),IGM(2),DRFTCO,LSRC,IELC,IGC
186.          DO 294 II = 1,70
187.          WRITE (IOUT,5290) II,VALM(II)
188.      294 CONTINUE
189.      C
190.      C *****
191.      C *
192.      C * COMPUTATION OF OBSERVED GRAVITY AND ANOMALIES
193.      C *
194.      C * BEGIN A GROUP OF MEASUREMENTS
195.      C *
196.      C *****
197.      C
198.      C * READ BASE GRAVITY
199.      C AND CONVERT INTO UNITS COMPATIBLE WITH BOTH SYSTEMS
200.      C
201.      300 CONTINUE
202.          READ (IIN,5320, END=910) BASEG(1),BASEG(2)
203.          READ (IIN,5330, END=910) DENSE
204.          BASG7=((BASEG(1)-977.)*1000.)*BASEG(2)
205.          IBASE=BASEG(1)
206.      C
207.      C * READ COUNTER READING CARDS FOR INDIVIDUAL STATIONS
208.      C THE FIRST CARD READ IS THE ONE FOR THE REFERENCE STATION
209.      C DRIFT IS COMPUTED FROM DATE ON THIS FIRST CARD.
210.      C ALL OTHER READINGS ARE REFERENCED TO THIS MEASUREMENT.
211.      C
212.          DO 890 I = 1,9000
213.          READ (IIN,5405, END=910)
214.          1 LSTAT,LDAY,M0,LYR,LTIME,CRN,LAT,RLATM,S0RN,L0NG,
215.          IRL0M,W0RE,ELEV,KTZ ,(DESC(IK),IK=1,31)
216.          IPCS=0
217.          CLS=.99
218.          H0NK=.99
219.          CR=CRN
220.          NCR=CRN
221.      C
222.      C * CHECK TO SEE IF HAVE NEW REFERENCE STATION OR CALL EXIT
223.      C (STMT #180 IS EXIT; #300 IS START OF NEW GROUP)
224.      C
225.          IF(NCR)500,500,420
226.          500 IF(LYR)415,180,415
227.          415 I=1
228.          GOT0 300
229.      C
230.      C * SEARCH TABLES FOR GRAVITY VALUE CORRESPONDING TO COUNTER READING
231.      C
232.          420 TUCR=CR*0.01
233.          LC=TUCR
234.          CI=LC
235.          C1=CI*100.
236.          C2=CR-CI
237.          RELM=VALM(LC)+(C2*.01*(VALM(LC+1)-VALM(LC)))
238.      C
239.      C * CALCULATE LAT AND L0N IN RADIANS AND IN DECIMAL DEGREES

```

CR2G1690  
CR2G1700  
CR2G1710  
CR2G1720  
CR2G1730  
CR2G1740  
CR2G1750  
CR2G1760  
CR2G1770  
CR2G1780  
CR2G1790  
CR2G1800  
CR2G1810  
CR2G1820  
CR2G1830  
CR2G1840  
CR2G1850  
CR2G1860  
CR2G1870  
CR2G1880  
CR2G1890  
CR2G1900  
CR2G1910  
CR2G1920  
CR2G1930  
CR2G1940  
CR2G1950  
CR2G1960  
CR2G1970  
CR2G1980  
CR2G1990  
CR2G2000  
CR2G2010  
CR2G2020  
CR2G2030  
CR2G2040  
CR2G2050  
CR2G2060  
CR2G2070  
CR2G2080  
CR2G2090  
CR2G2100  
CR2G2110  
CR2G2120  
CR2G2130  
CR2G2140  
CR2G2150  
CR2G2160  
CR2G2170  
CR2G2180  
CR2G2190  
CR2G2200  
CR2G2210  
CR2G2220  
CR2G2230  
CR2G2240  
CR2G2250  
CR2G2260  
CR2G2270

240.	C		CR2G2280
241.		RDEG=LAT	CR2G2290
242.		DEC=RLATM*1.666666E -2	CR2G2300
243.		RLAT=RDEG+DEC	CR2G2310
244.		DLAT = RLAT	CR2G2320
245.		RAD=RLAT*(1.7453293D-2)	CR2G2330
246.		RLAT=RAD	CR2G2340
247.	C		CR2G2350
248.		RDEG=LONG	CR2G2360
249.		DEC=RLDM*1.666666666D-2	CR2G2370
250.		RLONG=RDEG + DEC	CR2G2380
251.		DLON = RLONG	CR2G2390
252.		RLONG=RLONG*1.7453293 D-2	CR2G2400
253.	C		CR2G2410
254.	C	* IF KTZ IS EQUAL TO 99 MEANS HAVE NOT MADE OR LOOKED UP THE	CR2G2420
255.	C	TIME ZONE CORRECTION. THERE THE FOLLOWING CALCULATIONS	CR2G2430
256.	C	ARE NOT NEEDED BECAUSE WE CANNOT CALCULATE THE TIDAL OR	CR2G2440
257.	C	HONKLE CORRECTIONS WITHOUT IT	CR2G2450
258.	C		CR2G2460
259.		IF(KTZ.NE.99) GO TO 610	CR2G2470
260.		KGDA=LDAY	CR2G2480
261.		KGM0=M0	CR2G2490
262.		KGYR=LYR	CR2G2500
263.		KGHM=LTIME	CR2G2510
264.		ID=0	CR2G2520
265.		GO TO 630	CR2G2530
266.	C		CR2G2540
267.		610 CONTINUE	CR2G2550
268.		KTT=KTZ	CR2G2560
269.		CALL CHGMT(LDAY,M0,LYR,LTIME,KTT,KGDA,KGM0,KGYR,KGHM,NTZ)	CR2G2570
270.		CALL M2DY(KGYR,KGM0,KGDA,ID)	CR2G2580
271.		ADAY = KGHM	CR2G2590
272.		ADAY = ADAY / 2400.	CR2G2600
273.		ADAY = ADAY + FL0AT (ID)	CR2G2610
274.	C		CR2G2620
275.	C	NORTH LAT OR EAST LON IS POSITIVE	CR2G2630
276.	C	SOUTH LAT OR WEST LON IS NEGATIVE	CR2G2640
277.	C		CR2G2650
278.		IF (RLAT) 620,619,620	CR2G2660
279.	619	IF (RLONG) 620,630,620	CR2G2670
280.	620	CONTINUE	CR2G2680
281.	C		CR2G2690
282.		IF(S0RN.EQ.SOUTH) RLAT=-RLAT; DLAT=-DLAT	CR2G2700
283.		IF(W0RE.EQ.WEST) RLONG=-RLONG; DLON=-DLON	CR2G2710
284.	C		CR2G2720
285.	C	* CALCULATE CLS AND HONK VALUES	CR2G2730
286.	C		CR2G2740
287.		IHR=KGHM/100	CR2G2750
288.		IMIN=KGHM-IHR*100	CR2G2760
289.		CALL TIDAL(RLAT,RLONG,KGYR,ID,IHR,IMIN,CLS,HONK,DTD)	CR2G2770
290.		RELM=RELM+CLS+HONK	CR2G2780
291.	C		CR2G2790
292.		630 CONTINUE	CR2G2800
293.		IF(I-1)440,440,450	CR2G2810
294.	C		CR2G2820
295.	C	* NEW REFERENCE STATION (= FIRST CARD OF GROUP) PROCESSING	CR2G2830
296.	C		CR2G2840
297.		440 REF=RELM	CR2G2850
298.		IDA1=KGDA	CR2G2860
299.		IM01=KGM0	CR2G2870

```

300.      IYR1=KGYR
301.      ITM1=KGMM
302.      LSTA=LSTAT
303.      IPCS=1
304.      LCNT = 0
305.      IPAGE = 1
306.
307. C * CALCULATE DRIFT
308. C
309. 450 DIFFR=RELM-REF
310.     CALL CDATE(IDA1,IMO1,IYR1,ITM1,KGDA,KGMB,KGYR,KGMM,TIMD)
311.     DRIFT=(TIMD/24.0)*DRFTC0
312.     G0BS7=BASG7+DIFFR-DRIFT
313.     IC0UNT=IC0UNT+1
314.
315. C * PUT G0BS7 + 977000. INTO OUTPUT UNITS
316. C
317.     IC0N=G0BS7/1000.
318.     TEMP=IC0N*1000
319.     G0BS=G0BS7-TEMP
320.     IG=IC0N*977
321.
322. C * COMPUTE FREE-AIR AND BOUGUER ANOMALIES
323. C
324.     X=2.*RAD
325.     C2R=C0S(X)
326.     FELEV=((0.30855+0.00022*C2R)*ELEV)-(((ELEV*0.001)**2)*0.072)
327.     RA1=RAD
328.     GFREE=G0BS7-G167F(RA1)+FELEV
329.     BELEV=0.04185*DENSE*ELEV
330.     G00UG=GFREE-BELEV
331.
332. C * CHECK TO SEE IF LAT AND L0N = 0
333. C   OR IF ELEV = 0
334. C   IF THEY DO SET GFREE AND G00UG EQUAL TO 999.0
335. C
336.     IF(LAT)2050,2049,2050
337. 2049 IF(RLATM) 2050,2051,2050
338. 2051 IF(L0NG) 2050,2052,2050
339. 2052 IF(RL0M) 2050,2053,2050
340. C
341. 2050 IF (ELEV) 2054,2053,2054
342. 2053 GFREE=999.0
343. 2053 G00UG=999.0
344. 2054 CONTINUE
345. C
346.     NSTATN=LSTAT
347. C
348. C
349. C * COMPUTE GSUM SORT KEY FIELDS
350. C
351.     PLAT = DLAT + 90.
352.     LTKEY = PLAT
353.     PL0N = DL0N + 180.
354.     LGKEY = PL0N
355.     IAKY = 0
356. C
357. C *****
358. C *
359. C * OUTPUT THE DESIRED INFORMATION

```

```

CR2G2880
CR2G2890
CR2G2900
CR2G2910
CR2G2920
CR2G2930
CR2G2940
CR2G2950
CR2G2960
CR2G2970
CR2G2980
CR2G2990
CR2G3000
CR2G3010
CR2G3020
CR2G3030
CR2G3040
CR2G3050
CR2G3060
CR2G3070
CR2G3080
CR2G3090
CR2G3100
CR2G3110
CR2G3120
CR2G3130
CR2G3140
CR2G3150
CR2G3170
CR2G3180
CR2G3190
CR2G3200
CR2G3210
CR2G3220
CR2G3230
CR2G3240
CR2G3250
CR2G3260
CR2G3270
CR2G3280
CR2G3290
CR2G3300
CR2G3310
CR2G3320
CR2G3330
CR2G3340
CR2G3350
CR2G3360
CR2G3370
CR2G3380
CR2G3390
CR2G3400
CR2G3410
CR2G3420
CR2G3430
CR2G3440
CR2G3450
CR2G3460
CR2G3470

```

360.	C *	CR2G3480
361.	C *****	CR2G3490
362.	C	CR2G3500
363.	C * PUNCH OUTPUT FOR INPUT TO STATION DESCRIPTION PROGRAM	CR2G3510
364.	C 1PCS PREVENTS US FROM PUNCHING THE REFERENCE STATION WHEN	CR2G3520
365.	C IT IS THE FIRST CARD OUT	CR2G3530
366.	C	CR2G3540
367.	IF (1PCS.EQ.1) GOT0 835	CR2G3550
368.	IF (ISW(2)) 835,825,835	CR2G3560
369.	825 WRITE (JTAPE,5825) LYS,M0,LDAY,LTIME,LSTAT,IGM(1),IGM(2),	CR2G3570
370.	1 IG,G0BS,LSTA,IBASE,BASEG(2),RLAT,SORN,RLONG,WORE,ELEV	CR2G3580
371.	835 CONTINUE	CR2G3590
372.	C	CR2G3600
373.	C * LISTING OF CALCULATED VALUES	CR2G3610
374.	C	CR2G3620
375.	IF (ISW(1)) 869,851,869	CR2G3630
376.	851 IF (LCNT) 852,852,860	CR2G3640
377.	C	CR2G3650
378.	C PRINT PAGE HEADING	CR2G3660
379.	C	CR2G3670
380.	852 WRITE (IOUT,5001)	CR2G3680
381.	IF (IPAGE = 1) 853,853,854	CR2G3690
382.	853 WRITE (IOUT,5853)	CR2G3700
383.	854 CONTINUE	CR2G3710
384.	WRITE (IOUT,5200) IPAGE,KDATE	CR2G3720
385.	WRITE (IOUT,5855) LSTA,IDA1,IM01,IYR1,ITM1,IGM(1),IGM(2),LSRC	CR2G3730
386.	WRITE (IOUT,5856) IBASE,BASEG(2),REF,DENSE,DRFTC8	CR2G3740
387.	WRITE (IOUT,5858)	CR2G3750
388.	IPAGE = IPAGE + 1	CR2G3760
389.	LCNT = 39	CR2G3770
390.	C	CR2G3780
391.	860 WRITE (IOUT,5860)	CR2G3790
392.	1 LSTAT,LDAY,M0,LYR,LTIME,KTZ,LAT,RLATM,SORN,CR,	CR2G3800
393.	2 GFREE,CLS,DIFFR,DESC	CR2G3810
394.	WRITE (IOUT,5862)	CR2G3820
395.	1 IG,G0BS,KGDA,KGM0,KGYR,KGHM,ELEV,LONG,RL0M,WORE,RELM,	CR2G3830
396.	2 GBOUG,H0NK,DRIFT,TIMD,ADAY	CR2G3840
397.	LCNT = LCNT + 3	CR2G3850
398.	869 CONTINUE	CR2G3860
399.	C	CR2G3870
400.	C * OUTPUT AT GSUM FORMAT TO KTAPE	CR2G3880
401.	C FORMAT FORWARD CODE = 3	CR2G3890
402.	C	CR2G3900
403.	IF (ISW(4)) 889,871,889	CR2G3910
404.	871 WRITE (KTAPE,5871) IREC,LSRC,KGDA,KGM0,KGYR,KGHM,DLAT,DL0N,ELEV,	CR2G3920
405.	1 IG,G0BS,IDEF,GFREE,GBOUG,TC0RR,IELC,IGC,RFA,IREGC,	CR2G3930
406.	2 IFFC,CLS,H0NK,CRN,(DESC(11),11=1,16),NSTATN,IFBC,	CR2G3940
407.	3 LYKEY,LGKEY,IAKEY	CR2G3950
408.	889 CONTINUE	CR2G3960
409.	C	CR2G3970
410.	C * STMT. #890 IS THE END OF THE READ 06-000P	CR2G3980
411.	C	CR2G3990
412.	890 CONTINUE	CR2G4000
413.	C	CR2G4010
414.	C *****	CR2G4020
415.	C *	CR2G4030
416.	C * END OF JOB	CR2G4040
417.	C *	CR2G4050
418.	C *****	CR2G4060
419.	C	CR2G4070

```

420. 180 CONTINUE
421. 181 CONTINUE
422. 910 CONTINUE
423. WRITE (IOUT,5001)
424. OUTPUT ICOUNT
425. IF (ISW(4).GT.0) GO TO 24
426. ENDFILE KTAPE
427. 5950 WRITE (IOUT,5950)
428. REWIND KTAPE
429. 24 CONTINUE
430. IPAGE = 0
431. WRITE (IOUT,5200) IPAGE,KDATE
432. STOP
433. C
434. C *****
435. C *
436. C * FORMATS
437. C *
438. C *****
439. C
440. 5001 FORMAT (1H1)
441. 5150 FORMAT (12,F7.2)
442. 5200 FORMAT (T2'PAGE',I4,T35'DATE OF RUN = ',4A4)
443. 5230 FORMAT (2A4,2X,F10.5,I5,I5,I5)
444. 5285 FORMAT (' G=METER = ',2A4' DRFTC8 = ',F10.5,5X'SOURCE CODE = ',I4
445. 1 ' ELEV CODE = ',I4,5X'G METER CODE = ',I4)
446. 5290 FORMAT (' TABLE: ',5(I2,1,1,F7.2))
447. 5320 FORMAT (F3.0,F6.2)
448. 5330 FORMAT (F4.2)
449. 5405 FORMAT (I4,3I2,I4,F8.3,I2,F5.2,A1,I3,F5.2,A1,F7.1,I3,31A1)
450. 5825 FORMAT (3I2,I4,1X,I4,2A4,I3,F6.2,1X,I4,I3,F6.2,2(F9.6,A1),F7.1)
451. 5855 FORMAT ('/ REFERENCE STATION = ',I5,10X'READING OF ',3(I2,1,1)I4,
452. 1 5X'METER = ',2A4,5X'SOURCE CODE = ',I4)
453. 5853 FORMAT (T2'*** NEW REFERENCE STATION ***'/
454. 1 T2'*****')
455. 5856 FORMAT (' REFERENCE GRAVITY = ',I3,F6.2,7X'REL MGAL = ',F11.3,
456. 1 5X'DENSE = ',F5.2,9X'DRFTC8 = ',F6.4,/)
457. 5858 FORMAT (' STATION',10X'DATE',10X'TZ',4X'LATITUDE',2X'CTR RDNG'
458. 1 4X'GFREE',3X'CLS', 5X'GDIFF',2X'DESCRIPTION',/
459. 2 ' OBS GRAV',5X'GMT DATE',8X'ELEV',3X'LONGITUDE',2X'REL MGAL'
460. 3 4X'GBOUG',2X'WONK',2X'ACUM DFT',7X'TDIFF',2X'DA-OF-YR',/)
461. 5860 FORMAT (' ',I4,7X,3(I2,1,1)I4,6X,I3,2X,I3,1X,F5.2,A1,3X,
462. 1 2(F7.2,2X),F4.2,1X,F9.3,2X,31A1)
463. 5862 FORMAT (' ',I3,F6.2,2X,3(I2,1,1)I4,2X,F7.1,2X,I3,1X,F5.2,A1,3X,
464. 1 2(F7.2,2X)F4.2,2X,F8.2,2X,F10.2,2X,F8.4,
465. 2 /)
466. 5871 FORMAT (I1,I4,3I2,I4,2F9.4,F7.2,
467. 1 I3,F6.2,I5,2F6.1,F4.1,2I2,F6.1,I1,
468. 2 I2,2F4.2,F7.2,16A1,I4,I2,
469. 3 2I3,I2)
470. 5950 FORMAT(' WROTE END OF FILE')
471. C
472. END

```

CR2G4080  
 CR2G4090  
 CR2G4100  
 CR2G4110  
 CR2G4120  
 CR2G4130  
 CR2G4140  
 CR2G4150  
 CR2G4160  
 CR2G4170  
 CR2G4180  
 CR2G4190  
 CR2G4200  
 CR2G4210  
 CR2G4220  
 CR2G4230  
 CR2G4240  
 CR2G4250  
 CR2G4260  
 CR2G4270  
 CR2G4280  
 CR2G4290  
 CR2G4300  
  
 CR2G4330  
 CR2G4340  
 CR2G4350  
 CR2G4360  
 CR2G4370  
 CR2G4380  
 CR2G4390  
 CR2G4400  
 CR2G4410  
 CR2G4420  
 CR2G4430  
 CR2G4440  
 CR2G4450  
 CR2G4460  
 CR2G4470  
 CR2G4480  
  
 CR2G4500  
 CR2G4510  
 CR2G4520  
 CR2G4530  
 CR2G4540  
 CR2G4550  
 CR2G4560  
 CR2G4570  
 CR2G4580  
 CR2G4590

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
ADAY	R	SCALR	000BE V	1	ASOFT	R	SCALR	0007E V	1	BASEG	R	SCALR	0004A V	2	ADAY	R	SCALR	000BE V	1
BASG7	R	SCALR	00098 V	1	BELEV	R	SCALR	000D5 V	1	CDATE	R	SCALR	000AB V	1	BASG7	R	SCALR	00098 V	1
CHONT	R	SPRGG	00000 V	1	CI	R	SCALR	000B1 V	1	CLS	R	SCALR	000A0 V	1	CHONT	R	SPRGG	00000 V	1
COS	R	SPRGG	00000 V	1	C2R	R	SCALR	000AD V	1	CRN	R	SCALR	00070 V	2	COS	R	SPRGG	00000 V	1
C2	R	SCALR	00082 V	1	DESC	R	SCALR	000D1 V	1	DEC	R	SCALR	000C8 V	1	C2	R	SCALR	00082 V	1
DENSE	R	SCALR	00097 V	1	DLO	R	SCALR	0004C V	31	DIFFR	R	SCALR	00092 V	1	DENSE	R	SCALR	00097 V	1
DIAT	R	SCALR	000B5 V	1	DTON	R	SCALR	0004C V	1	DRFTCO	R	SCALR	000A7 V	1	DIAT	R	SCALR	000B5 V	1
DRIFT	R	SCALR	000CA V	1	DTD	R	SCALR	0004C V	2	ELEV	R	SCALR	000D6 V	1	DRIFT	R	SCALR	000CA V	1
FELEV	R	SCALR	00002 V	1	FLGAT	R	SPRGG	0004C V	1	GBUG	R	SCALR	000CE V	1	FELEV	R	SCALR	00002 V	1
GFREE	R	SCALR	00004 V	1	GI67F	R	SCALR	0004C V	1	GBS	R	SCALR	0009A V	1	GFREE	R	SCALR	00004 V	1
GBS7	R	SCALR	000CB V	1	HONK	R	SCALR	0004C V	1	IBLNC	R	SCALR	0007A V	1	GBS7	R	SCALR	000CB V	1
IAKEY	R	SCALR	0000C V	1	IBASE	R	SCALR	00099 V	1	ID	R	SCALR	000BB V	1	IAKEY	R	SCALR	0000C V	1
ICGN	R	SCALR	000CC V	1	ICOUNT	R	SCALR	00099 V	1	IELC	R	SCALR	00094 V	1	ICGN	R	SCALR	000CC V	1
IDA1	R	SCALR	000C2 V	1	IDEP	R	SCALR	00094 V	1	IG	R	SCALR	000CF V	1	IDA1	R	SCALR	000C2 V	1
IFBC	R	SCALR	0008D V	1	IFPC	R	SCALR	0008C V	1	IG18	R	SCALR	0007C V	1	IFBC	R	SCALR	0008D V	1
IGC	R	SCALR	00095 V	1	IGH	R	SCALR	0008B V	1	IIN	R	SCALR	0007F V	1	IGC	R	SCALR	00095 V	1
IKR	R	SCALR	0008F V	1	ILR	R	SCALR	00096 V	1	IIN	R	SCALR	0007F V	1	IKR	R	SCALR	0008F V	1
IK	R	SCALR	000A9 V	1	ILR	R	SCALR	00096 V	1	IBUT	R	SCALR	000C0 V	1	IK	R	SCALR	000A9 V	1
IM01	R	SCALR	000C3 V	1	INN	R	SCALR	00091 V	1	IBUT	R	SCALR	000C0 V	1	IM01	R	SCALR	000C3 V	1
IPAGE	R	SCALR	00084 V	1	IPCS	R	SCALR	000AA V	1	IREC	R	SCALR	00085 V	1	IPAGE	R	SCALR	00084 V	1
IREGC	R	SCALR	00088 V	1	ISH	R	SCALR	00088 V	1	ITM1	R	SCALR	000C5 V	1	IREGC	R	SCALR	00088 V	1
IYR1	R	SCALR	00088 V	1	J	R	SCALR	0008F V	1	JTAP	R	SCALR	00081 V	1	IYR1	R	SCALR	00088 V	1
K	R	SCALR	00084 V	1	KDATE	R	SCALR	0008F V	1	KGDA	R	SCALR	000B7 V	1	K	R	SCALR	00084 V	1
KGMH	R	SCALR	0008A V	1	KGNB	R	SCALR	0008B V	4	KGYR	R	SCALR	000B9 V	1	KGMH	R	SCALR	0008A V	1
KYAP	R	SCALR	0008A V	1	KTY	R	SCALR	0008C V	1	KZ	R	SCALR	000A8 V	1	KYAP	R	SCALR	0008A V	1
LAT	R	SCALR	00082 V	1	LC	R	SCALR	00080 V	1	LCNT	R	SCALR	000C7 V	1	LAT	R	SCALR	00082 V	1
LDAY	R	SCALR	000A1 V	1	LELC	R	SCALR	0008A V	1	LGC	R	SCALR	000B3 V	1	LDAY	R	SCALR	000A1 V	1
LGKEY	R	SCALR	0009C V	1	LONG	R	SCALR	000A4 V	1	LSRC	R	SCALR	000B3 V	1	LGKEY	R	SCALR	0009C V	1
LRCD	R	SCALR	000DB V	1	LSTA	R	SCALR	000A4 V	1	LSTAT	R	SCALR	000B3 V	1	LRCD	R	SCALR	000DB V	1
LTIME	R	SCALR	0007D V	1	LTKY	R	SCALR	000C6 V	1	LYR	R	SCALR	0009B V	1	LTIME	R	SCALR	0007D V	1
HO	R	SCALR	0009F V	1	M2DY	R	SCALR	000D9 V	1	NCR	R	SCALR	0009E V	1	HO	R	SCALR	0009F V	1
NSATN	R	SCALR	0009D V	1	NTZ	R	SCALR	000D9 V	1	PLAT	R	SCALR	000A4 V	1	NSATN	R	SCALR	0009D V	1
PLGN	R	SCALR	000D7 V	1	RAD	R	SCALR	000D9 V	1	RA1	R	SCALR	000D3 V	1	PLGN	R	SCALR	000D7 V	1
REG	R	SCALR	000DA V	1	REF	R	SCALR	00074 V	2	RELH	R	SCALR	000B3 V	1	REG	R	SCALR	000DA V	1
RFA	R	SCALR	00084 V	1	RLAT	R	SCALR	000C1 V	1	RELH	R	SCALR	000B3 V	1	RFA	R	SCALR	00084 V	1
RLGN	R	SCALR	00087 V	1	RLONG	R	SCALR	00072 V	2	SBRN	R	SCALR	000A2 V	1	RLGN	R	SCALR	00087 V	1
SAUTH	R	SCALR	000A5 V	1	TABLE	R	SCALR	00076 V	2	TCRR	R	SCALR	000A3 V	1	SAUTH	R	SCALR	000A5 V	1
TEMP	R	SCALR	00079 V	1	TIDAL	R	SCALR	00090 V	1	TIDH	R	SCALR	000A3 V	1	TEMP	R	SCALR	00079 V	1
TODAY	R	SCALR	000CD V	1	TUCR	R	SCALR	00090 V	1	VALH	R	SCALR	000A3 V	1	TODAY	R	SCALR	000CD V	1
WEST	R	SCALR	00078 V	1	WORE	R	SCALR	000AF V	1	X	R	SCALR	000A4 V	70	WEST	R	SCALR	00078 V	1

HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL
002C6	24	000A0	300	0009C	294	0004C	210	002AC	181
000E9	415	002C0	595	000E7	500	00195	450	00183	440
00134	610	00218	835	00203	825	00180	630	00157	620
00210	851	00277	869	0024E	869	0022A	854	00226	853
0027C	871	001E5	2050	001DF	2049	002AC	910	002A8	890
001E1	2051	002D5	5150	002D2	5001	001EB	2054	001E7	2053
00209	5200	0030F	5330	0030B	5320	0030F	5290	002EA	5285
00312	5405	5858	5858	0035A	5856	0032F	5855	00347	5853
003A6	5860	00372	5858	0035A	5856	0032F	5855	00347	5853
						003ED	5950	00305	5871

## LOCAL VARIABLES (221 WORDS):

00000 KDATE	00004 VALM	0004A BASEG	0004C DESC	0006B IGM	0006E DTD
00070 DEC	00072 RLAT	00074 RAD	00076 RLNG	00078 WEST	00079 SOUTH
0007A IBLNK	00078 ILR	0007C IG18	0007D LSRCD	0007E ASDFT	0007F IIN
00080 IOUT	00081 JTAPE	00082 KTAPE	00083 ICOUNT	00084 IPAGE	00085 IREC
00086 IDEP	00087 RFA	00088 IREGC	00089 TCERR	0008A LELC	0008B LGC
0008C IFFC	0008D IFBC	0008E K	0008F J	00090 TABLE	00091 INN
00092 DRFTCB	00093 LSRC	00094 IELC	00095 IGC	00096 IJ	00097 DENSE
00098 BASG7	00099 IBASE	0009A I	0009B LSTAT	0009C LDAY	0009D M8
0009E LYR	0009F LTIME	000A0 CRN	000A1 LAT	000A2 RLATH	000A3 SORN
000A4 LONG	000A5 RLGM	000A6 WORE	000A7 ELEV	000A8 KYZ	000A9 IK
000AA IPCS	000AB CLS	000AC H8NK	000AD CR	000AE NCR	000AF TUCR
000B0 LC	000B1 CI	000B2 C2	000B3 RELM	000B4 RDEG	000B5 DLAT
000B6 DLEN	000B7 KGDA	000B8 KGM8	000B9 KGYR	000BA KGHM	000BB ID
000BC KTY	000BD NTZ	000BE ADAY	000BF IHR	000C0 IMIN	000C1 REF
000C2 IDA1	000C3 IM81	000C4 IYR1	000C5 ITM1	000C6 LSTA	000C7 LCNT
000C8 DIFFR	000C9 TIMD	000CA DRIFT	000CB G8BS7	000CC IC8N	000CD TEMP
000CE G8BS	000CF IG	000D0 X	000D1 C2R	000D2 FELEV	000D3 RA1
000D4 GFREE	000D5 BELEV	000D6 G8UG	000D7 NSTATN	000D8 PLAT	000D9 LTKEY
000DA PLON	000DB LGKEY	000DC IKEY			

## BLANK COMMON (0 WORDS)

## INTRINSIC SUBPROGRAMS USED:

C8S      FLOAT

## EXTERNAL SUBPROGRAMS REQUIRED:

CDATE	CHGMT	G167F	ISM	M2DY	TIDAL	T8DAY	F1101
F1102	F1103	F1104	F1105	F1106	F1108	M1D8	M18C
9BCDRDEE	9BCDREAD	9BCDWRIT	9C8S	9DTRR	9ENDFILE	9ENDJOL	9INITIAL
910DATA	910LUSA	91TRR	9PRINT	9REWIND	9RT81	9ST8P	

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC	HEX
WORDS	WORDS
-----	-----
GENERATED CODE: 1014	003F6
CONSTANTS: 30	0001E
LOCAL VARIABLES: 221	00000
TEMPS: 1	00001
-----	-----
TOTAL PROGRAM: 1266	004F2

*Compiled 1 Apr 1972*

```

1.  C  PROGRAM CRWT3
2.  C  VERSION OF 7 MARCH 1972, TO USE PINOT
3.  C
4.  C  DIMENSION TAB(100),IDESC(6),VEL(8),THICK(8),X(8)
5.  C  DIMENSION N0W(4)
6.  C
7.  C  PROGRAM CRWT3, CALCULATES PRESSURE AT BASE OF CRUSTAL
8.  C  COLUMN (KG/CM2)
9.  C
10. C  SSW(0) UP TO LIST INTERMEDIATE VALUES FOR TESTING
11. C  SSW(26) UP TO SET JTAPE = 108 AND IREC1 = 0
12. C  SSW(32) UP TO READ SPFMT DATA ON TWO CARDS
13. C  SSW(33) UP TO WRITE SPFMT DATA ON TWO CARDS
14. C
15. C  USES SUBROUTINES EVIL, ISW, STAT
16. C
17. C
18. C  *****
19. C  ITAPE = URN FOR SEISMIC DATA INPUT
20. C  JTAPE = URN FOR DATA OUTPUT
21. C  ITAPE = 1
22. C  JTAPE = 2
23. C  *****
24. C
25. C  IIN = 105
26. C  IIOUT = 108
27. C  NOUT=0
28. C
29. C  PRINT DATE AND TIME OF JOB ON HEADING
30. C  CALL TODAY(N0W)
31. C  WRITE(IIOUT,13) N0W
32. C  13 FORMAT(1X,4A4)
33. C  INIT = ISW(-2)
34. C  CALL STAT
35. C  K9 = 149
36. C  ISTAB=0
37. C  WRITE(IIOUT,600)
38. C  600 FORMAT ( / 'PROGRAM CRWT3,VERSION OF 7 MARCH,1972' //)
39. C
40. C  KK=0
41. C  CALL PINOT(ITAPE,JTAPE,KK,ISTA,KEY,LAT,LATM,KNS,
42. C  1 LONG,L0M,KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,
43. C  2 DINE,STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTW)
44. C
45. C
46. C  DCOMP = DEPTH OF COMPENSATION IN KM.
47. C  ICTAB = 0 FOR NAPE DRAKE, = 1 FOR WOOLARD DENSITY TABLE
48. C  READ (IIN, 2) ICTAB, DCOMP
49. C  2 FORMAT (15, F10.0)
50. C  OUTPUT ICTAB, DCOMP
51. C  READ IN 10 VALUES PER CARD
52. C  READ (IIN, 3) TAB

```

```

53.      3      FORMAT (10F8.3)
54.      C      100 VALUES ENTERED
55.      C
56.      C READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
57.      10      CONTINUE
58.          KK=1
59.          CALL      PRINT(ITAPE,OTAPE,KK,ISTA,KEY,LAT,LATM,KNS,
60.      1  LONG,LBM,KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,DESC,
61.      2  DINE,STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTW)
62.          IF(KK-9)120,540,120
63.      120      CONTINUE
64.          IF(ISA(0))16,18,16
65.      16      WRITE(IIOUT,17) ISTA,KEY,VMANT,ELEV,N1,N2,N3,N4
66.      17      FORMAT('READ',3X,15,13,F5.1,F7.1,3X,4I1)
67.      C CHECKING IF KEY = 9
68.      18      IF(KEY-K9)20,10,20
69.      20      CONTINUE
70.          ELEV=NELEV
71.          ELEV=ELEV*0.01
72.          VMANT=(FLBAT(IMANT))*0.1
73.          NCT=8
74.      45      IF(IMANT)50,850,50
75.      50      IF(N1-2) 70,60,70
76.      C SEA SEISMIC PROFILE
77.      60      DINE = ELEV
78.          WGT = 1.03*ELEV*100.0
79.          GO TO 80
80.      C LAND SEISMIC PROFILE
81.      70      DINE = -ELEV
82.          WGT = 0.0
83.      80      WGTW = WGT
84.          IF(ISA(0))81,83,81
85.      81      WRITE(IIOUT,82)DINE,WGT
86.      82      FORMAT('DINE=',F4.2,4X,'WGT=',F10.2)
87.      83      SX = 0.0
88.          STHIK = 0.0
89.          DO 86 J=1,NCT
90.          KK = VEL(J)*10.0
91.          DENS = TAB(KK)
92.          WGT = WGT+(DENS*THICK(J)*100.0)
93.          DINE = DINE + THICK(J)
94.          X(J) = VEL(J)*THICK(J)
95.          SX = SX + X(J)
96.          STHIK = STHIK + THICK(J)
97.          IF(ISA(0))84,86,84
98.      84      WRITE(IIOUT,85)J,KK,THICK(J),DENS,WGT,DINE,
99.      1  X(J),SX,STHIK
100.      85      FORMAT('DB LOOP',2X,I2,14,2X,F4.1,2X,F4.2,
101.      1  2X,F10.2,2X,F4.1,2X,F6.2,2X,F6.2,2X,F4.1)
102.      86      CONTINUE
103.          CRVEL = SX/STHIK
104.          KK = CRVEL*10.0+0.5
105.          CRDEN = TAB(KK)

```

```

106.      KK = VMANT*10.0+0.5
107.      DENS = TAB(KK)
108.      A = CRDEN*STHIK*100.0
109.      B = DENS*(DCOMP-DINE)*100.0
110.      IF (ISW(0)) 90,95,90
111.      90  WRITE(IIOUT,92)CRDEN,DENS,A,B
112.      92  FORMAT('CRDEN=',F5.2,3X,'DENS=',F5.2,4X,
113.      1    'A=',F10.2,3X,'B=',F10.2)
114.      95  WGT = WGT + B
115.      AVWGT = WATW + A + B
116.      C
117.      C  SETTING UP FOR PROPER OUTPUT
118.      IF (ICTAB) 810,820,810
119.      810  CRVW = CRVEL
120.      WGTW = WGT
121.      AVWTW = AVWGT
122.      GO TO 850
123.      820  CRVN = CRVEL
124.      WGTN = WGT
125.      AVWTN = AVWGT
126.      C  OUTPUT RESULTS
127.      850  CONTINUE
128.      KK=-2
129.      CALL      PINDT(ITAPE,UTAPE,KK,ISTA,KEY,LAT,LATM,KNS,
130.      1  LONG,LON,KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,DESC,
131.      2  DINE,STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTW)
132.      NOUT=NOUT+1
133.      GO TO 10
134.      540  WRITE(IIOUT,545)NOUT
135.      545  FORMAT('END FOUND ON INPUT TAPE ',I10)
136.      NOUT=0
137.      END FILE UTAPE
138.      999  CALL EXIT
139.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	000B3 V	1	AVGHT	R	SCALR	000B5 V	1	AVWTN	R	SCALR	000A3 V	1	AVWTN	R	SCALR	000A3 V	1
AVMTN	R	SCALR	000A6 V	1	B	R	SCALR	000B4 V	1	CRDEN	R	SCALR	000B2 V	1	CRDEN	R	SCALR	000B2 V	1
CRVEL	R	SCALR	000B1 V	1	CRVN	R	SCALR	000A1 V	1	CRVW	R	SCALR	000A4 V	1	CRVW	R	SCALR	000A4 V	1
DC8MP	R	SCALR	000A8 V	1	DENS	R	SCALR	000B0 V	1	DINE	R	SCALR	0009F V	1	DINE	R	SCALR	0009F V	1
ELEV	R	SCALR	000AA V	1	EXIT	R	SPR8G	EXTER:	1	FLOAT	R	SPR8G	INTRIN	1	FLOAT	R	SPR8G	INTRIN	1
ICTAB	I	SCALR	000A7 V	1	IDESC	I	ARRAY	00064 V	6	IIN	I	SCALR	00088 V	1	IIN	I	SCALR	00088 V	1
IIBUT	I	SCALR	00089 V	1	IMANT	I	SCALR	00097 V	1	INIT	I	SCALR	0008B V	1	INIT	I	SCALR	0008B V	1
ISTA	I	SCALR	0008F V	1	ISTAB	I	SCALR	0008D V	1	ISM	I	SPR8G	EXTERN	1	ISM	I	SPR8G	EXTERN	1
ITAPE	I	SCALR	00086 V	1	IYR	I	SCALR	0009E V	1	J	I	SCALR	000AF V	1	J	I	SCALR	000AF V	1
JTAPE	I	SCALR	00087 V	1	KEW	I	SCALR	00036 V	1	KEY	I	SCALR	00090 V	1	KEY	I	SCALR	00090 V	1
KK	I	SCALR	0008E V	1	KNS	I	SCALR	00093 V	1	K9	I	SCALR	0008C V	1	K9	I	SCALR	0008C V	1
LAT	I	SCALR	00091 V	1	LATH	I	SCALR	00032 V	1	L8M	I	SCALR	00095 V	1	L8M	I	SCALR	00095 V	1
LONG	I	SCALR	00094 V	1	MET	I	SCALR	0009D V	1	NCT	I	SCALR	000AB V	1	NCT	I	SCALR	000AB V	1
NELEV	I	SCALR	00098 V	1	NBUT	I	SCALR	0008A V	1	N8W	I	ARRAY	00082 V	4	N8W	I	ARRAY	00082 V	4
N1	I	SCALR	00099 V	1	N2	I	SCALR	0009A V	1	N3	I	SCALR	0009B V	1	N3	I	SCALR	0009B V	1
N4	I	SCALR	0009C V	1	P1NBT	I	SPR8G	EXTERN	1	STAT	R	SPR8G	EXTERN	100	STAT	R	SPR8G	EXTERN	100
STH1K	R	SCALR	000A0 V	1	SX	R	SCALR	000AE V	1	TAB	R	ARRAY	0006A V	8	TAB	R	ARRAY	0006A V	8
THICK	R	ARRAY	00072 V	8	T8DAY	R	SPR8G	EXTERN	1	VEL	R	ARRAY	0006A V	1	VEL	R	ARRAY	0006A V	1
VMANT	R	SCALR	000A9 V	1	WATW	R	SCALR	000AD V	1	WGT	R	SCALR	000AC V	1	WGT	R	SCALR	000AC V	1
WGTN	R	SCALR	000A2 V	1	WGTW	R	SCALR	000AD V	1	X	R	ARRAY	0007A V	8	X	R	ARRAY	0007A V	8

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
2	0005A	3	00072	10	00075	13	00014	16	0009F
18	000B5	20	000B8	45	000C4	50	000C6	60	000C9
80	000D5	81	000DC	82	000E2	83	000EB	84	0010E
86	00134	90	0015A	92	00162	95	00172	120	0009A
545	001B1	600	00025	810	0017B	820	00182	850	00188

## LOCAL VARIABLES (182 WORDS):

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
00000	TAB		00064	IDESC	00072	THICK		00072	1	0007A	X		0007A	1	00082	NGW		00082	1
00086	ITAPE		00087	JTAPE	00085	IIBUT		00085	1	0008A	NBUT		0008A	1	00088	INIT		00088	1
0008C	K9		0008D	ISTA9	0008F	ISTA		0008F	1	00090	KEY		00090	1	00091	LAT		00091	1
00092	LATM		00093	KNS	00095	L8M		00095	1	00096	KEW		00096	1	00097	IMANT		00097	1
00098	NELEV		00099	N1	0009B	N3		0009B	1	0009C	N4		0009C	1	0009D	MET		0009D	1
0009E	IYR		0009F	DINE	000A1	CRVN		000A1	1	000A2	WGTN		000A2	1	000A3	AVWTN		000A3	1
000A4	CRVW		000A5	WGTW	000A7	ICTAB		000A7	1	000A8	DC8MP		000A8	1	000A9	VMANT		000A9	1
000AA	ELEV		000AB	NCT	000AD	WATW		000AD	1	000AE	SX		000AE	1	000AF	J		000AF	1
000AC	DENS		000B1	CRVEL	000B2	CRDEN		000B2	1	000B3	A		000B3	1	000B4	B		000B4	1

BLANK CAMPON (0 WORDS)

## INTRINSIC SUBPROGRAMS USED:

FLOAT

## EXTERNAL SUBPROGRAMS REQUIRED:

EXIT	ISN	PINOT	STAT	TODAY	F:101	F:102	F:103
F:104	F:105	F:106	F:108	M:D0	M:8C	9BCDREAD	9BCDWRT
9ENDFILE	9ENDI0L	9INITIAL	9I0DATA	9I0LUSA	9IT0R	9PRINT	9RT0.
9STOP							

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS	
GENERATED CODE:	451	C01C3	(NO MEMORY PROTECTION)
CONSTANTS:	9	00009	
LOCAL VARIABLES:	182	000B6	
TEMPS:	0	00000	
TOTAL PROGRAM:	642	00282	

```

1.  C  PROGRAM DMABLK
2.  C  PROGRAM TO READ BLOCKED BY 50 DMA SOURCE TAPES
3.  C  AND CONVERT THEM TO BLOCKED BY 50 GSUM TAPES (IFFC=9)
4.  C  HYBRID OF PROGRAM DMA AND CONV67
5.  C  VERSION 20 AMY 75 TO REMOVE DOUBLE BUFFER
6.  C  VERSION OF 12 MAY 75 TO ZERO VARIABLES FOR CP-V
7.  C  VERSION ON 20 DEC 74 TO CORRECT TEST OF IELEV AND ZEROING OUT
8.  C  ELEV, IDEP, FA, BG, AND TC
9.  C
10. C  VERSION 30 SEPT 74 TO CORRECT SPELLING OF ISLAT IN READ
11. C  ORIGINAL VERSION 25 JULY 1974 BY G. GÖVE
12.  C  DIMENSION IA(35)
13.  C  DIMENSION IRUFIN(21,50),IBUFOT(32,50)
14.  C  DIMENSION IZ( 9),IW(35)
15.  C  DOUBLE PRECISION GGBS
16.  C  INTEGER ONE,TWO
17.  C  INTEGER THREE,FOUR,FIVE
18.  C  INTEGER SIX,SEVEN,EIGHT,NINE,DEE
19.  C  DATA ONE,TWO/'1 ','2 ' '/'
20.  C  DATA THREE,FOUR,FIVE/'3 ','4 ','5 ' '/'
21.  C  DATA SIX,SEVEN,EIGHT/'6 ','7 ','8 ' '/'
22.  C  DATA NINE,DEE/'9 ','D ' '/'
23.  C  JNEG=1H-
24.  C  OUTPUT 'DMABLK VERSION 20 MAY 75'
25.  C  OUTPUT 'ON DEC 12 74 IT WAS DISCOVERED THAT'
26.  C  OUTPUT 'THE LOGIC IN PROGRAM DMA DID NOT HANDLE'
27.  C  OUTPUT 'ELEVATION CODES OTHER THAN 1 AND 3 CORRECTLY'
28.  C  OUTPUT 'IT WAS ALSO DISCOVERED THAT DMABLK DID NOT ZERO'
29.  C  OUTPUT 'ELEV AND IDEP. TO FIX THIS A TEMPORARY '
30.  C  OUTPUT 'VERSION OF DMABLK WAS MADE'
31.  C  OUTPUT 'THIS VERSION WRITES 1 AND 3 IN GSUM '
32.  C  OUTPUT 'BUT JUST PASSES ALL OTHER ELEV CODES TO LTAPE'
33.  C  OUTPUT 'IN DMA FORMAT. ALL GSUM O/P IS GOOD'
34.  C  OUTPUT 'LEE GÖVE DEC 13 74'
35.  C  ITAPE=1
36.  C  JTAPE=2
37.  C  LTAPE=3
38.  C  NRECRD=0
39.  C  IODBAL=0
40.  C  IN=105
41.  C  IBUT=103
42.  C  IREC2=2
43.  C  ITAPE=1
44.  C  JTAPE=2
45.  C  KTAPE=108
46.  C  ICNT=0
47.  C  NIN=50
48.  C  NOUT=0
49.  C  IBUTSW=0
50.  C  DEGA=1.745329E-2
51.  C  KK=0
52.  C  KI=1
53.  C  KB=-2
54.  C  NREC=0
55.  C  A=0.0
56.  C  DLAT=0.0; DLONG=0.0
57.  C  HEIGHT=0.0
58.  C  IKEY=0
59.  C  IDEIF=0

```

```

60.      IE0D=0
61.      C
62.      C      INPUT STARTING SEQUENCE NUMBER FOR ID
63.      C
64.      READ(IN,8)NSEQ
65.      8  FORMAT(I10)
66.      OUTPUT NSEQ
67.      READ (IN,505) IS0RC
68.      505 FORMAT (I5)
69.      OUTPUT IS0RC
70.      KGDA=0
71.      KGMB=0
72.      KGYR=0
73.      KGHM=0
74.      IDIF=0
75.      KGDA0=0
76.      KGMB0=0
77.      KGYR0=0
78.      IELC=0
79.      IGC=0
80.      IREGC=0
81.      IFFC=9
82.      IFBC=0
83.      RFA=0.0
84.      C      BUFFER LOGIC FOR I/P
85.      C
86.      10 CONTINUE
87.      IF(NIN.LT.50) GO TO 90
88.      NIN=0
89.      CALL BUFF IN(ITAPE,0,IBUFIN(1,1),1050)
90.      15 CONTINUE
91.      CALL ICHECK(ITAPE,IKEY,NI)
92.      GO TO (20,50,30,40) IKEY
93.      20 OUTPUT 'WAITING FOR I/P'; IE0D=0
94.      GO TO 15
95.      30 OUTPUT 'END OF FILE ON ITAPE'; IE0D=1
96.      GO TO 50
97.      40 OUTPUT 'BUFFER IN ERROR'; IE0D=1
98.      GO TO 999
99.      50 CONTINUE
100.     C
101.     C      INPUT LOGIC
102.     C
103.     90 CONTINUE
104.     NIN=NIN+1
105.     IF(NI.EQ.1050) GO TO 95
106.     C      GOING TO EOF PROCESSING
107.     NINCHK=NIN*21
108.     IF(NINCHK.GT.NI) GO TO 999
109.     95 CONTINUE
110.     ELEV=0.0
111.     IDEP=0
112.     FA=999.0
113.     BG=999.0
114.     TC=99.9
115.     NRECRD=NRECRD+1
116.     DEC0DE(84,500,IBUFIN(1,NIN),ND)
117.     A      IGE0C,ISLAT,LAT,ALAT,ISLG,L0NG,AL0NG,IELEV,IELU,
118.     A      ELEV,DEPIN,G0BS,
119.     A      FA,BG,SOURCE,IBASE,IBR,ISEQ

```

```

120. 500  FORMAT(2X,I1,A1,I2,F4.2,1X,A1,I3,F4.2,1X,A1,I1,F7.1,1X,F5.1,1X,
121.      A   F6.2,1X,
122.      A   F5.1,1X,F5.1,3X,A4,1X,A4,A1,1X,A4,4X)
123.  C
124.  C   EDIT LOGIC
125.  C
126.  C
127.  C   TO OUTPUT RECORD SEQUENCE NUMBER IN STATION NUMBER FIELD
128.  C
129.      A=NSEQ
130.      KGYR=A*0.0001
131.      B=KGYR*10000
132.      KGHM=A-B
133. 507  KGHM=KGHM
134.      KGYR=KGYR
135.  C   GROUPING VARIABLES FOR OUTPUT UNDER ARRAY IA
136.      ENCODE(35,410,I2) SOURCE,IBASE,IBR,ISEQ,ELEV
137. 410  FORMAT(3X,A4,1X,A4,A1,1X,A4,1X,A1,15X)
138.      CALL UNPKBY(I2,IW,35)
139.      DB 420 J=1,35
140.      IA(J)=ISL(IW(J),24)
141. 420  CONTINUE
142.  C
143.  C   CALCULAT LAT, LONG, AND KEYS
144.  C
145.      DLAT=FLBAT(LAT)+(ALAT/60.0)
146.      IF (ISLAT.EQ. JNEG) DLAT=-DLAT
147.      DLONG=FLBAT(LONG)+ALONG/60.0
148.      IF (ISLG.EQ. JNEG) DLONG=-DLONG
149.      PLAT=DLAT+90.0; LTKEY=PLAT
150.      PLONG=DLONG+180.0; LGKEY=PLONG
151.  C   CHECKING GEOGRAPHIC COORDINATE CODE
152.      IF (IGERC.EQ.1) WRITE(IIOUT,510) SOURCE,ISEG ; GO TO 99
153. 510  FORMAT(' GEO CODE = 1, STOPPED PROCESSING AT ',A4,2X,A4)
154.      IF (IGERC.EQ.2) WRITE(IIOUT,520) SOURCE,ISEG ; GO TO 99
155. 520  FORMAT(' GEO CODE = 2, STOPPED PROCESSING AT ',A4,2X,A4)
156.  C   CONVERT ELEVATION TO METERS DEPENDING ON CODE
157.      IF (IELU.EQ.1) ELEV=ELEV/3.281
158.      IF (IELU.EQ.2) ELEV=ELEV/19.686
159.      IF (IELEV.EQ. ONE) GO TO 5215
160.      IF (IELEV.EQ. THREE) IDEP=ELEV; ELEV=0.0; GO TO 5215
161.      I9DBAL=I9DBAL+1
162.      WRITE(LTAPE,5555) (IBUFIN(KK,NIN),KK=1,21)
163. 5555 FORMAT(21A4)
164.      GO TO 10
165. 5215 CONTINUE
166.      IF (G8BS = 0.05) 521,521,524
167. 521  K977 = 0
168.      B8SG = 0.0
169.      GO TO 528
170. 524  G8BS=G8BS+976000.00
171.      CALL B8GD(K977,B8SG,G8BS,K8)
172.  C   CHECKING FOR VALID HEIGHT
173. 528  CALL ALTD(ELEV,IDEP,HEIGHT,KK)
174.      IF (KK-9) 550,530,550
175. 530  BG=999.0
176.  C   OUTPUT GSM RECORD
177. 550  CONTINUE
178.  C
179.  C   OUTPUT LOGIC

```

```

180. C
181. 300 CONTINUE
182.   NOUT=NOUT+1
183.   ENCODE(128,1001,IBUF8T(1,NOUT),ND) IREC2,ISBRC,KGDA,KGMS,
184.   1   KGYR,KGHM,DLAT,DLONG,ELEV,K977,0BSG,IDEF,FA,BG,TC,IELC,
185.   2   IGC,RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
186.   NREC=NREC+1
187.   NSEQ=NSEQ+1
188.   KGHM8=KGHM
189. 305 CONTINUE
190.   IF(NOUT.LT.50) GO TO 10
191. C
192. C   BUFFER LOGIC FOR O/P
193. C
194. 310 CONTINUE
195.   JKEY=ICHECK(JTAPE)
196.   GO TO (320,350,330,340) JKEY
197. 320 OUTPUT 'WAITING FOR O/P' ; IE8D=0
198.   GO TO 310
199. 330 OUTPUT END OF FILE JTAPE ; IE8D=1
200.   GO TO 999
201. 340 OUTPUT 'BUFF OUT ERROR' ; IE8D=1
202.   GO TO 999
203. 350 CONTINUE
204.   NOUT=0
205.   CALL BUFF OUT(JTAPE,0,IBUF8T(1,1),1600)
206.   GO TO 10
207. C
208. C   END OF JOB
209. C
210. 999 CONTINUE
211. 910 CONTINUE
212.   JKEY=ICHECK(JTAPE)
213.   GO TO (920,950,930,940) JKEY
214. 920 OUTPUT 'WAITING FOR O/P' ; IE8D=0
215.   GO TO 910
216. 930 OUTPUT 'BAD JKEY' ; IE8D=1
217.   GO TO 960
218. 940 OUTPUT 'BUFF OUT ERROR' ; IE8D=1
219.   GO TO 960
220. 950 CONTINUE
221.   JWDS=NOUT*32
222.   CALL BUFF OUT(JTAPE,0,IBUF8T(1,1),JWDS)
223. 960 CONTINUE
224. 99 CONTINUE
225.   END FILE LTAPE
226.   END FILE JTAPE
227.   NSEQ=NSEQ+1
228.   WRITE(10UT,1090)NREC,NSEQ
229. 1090 FORMAT('END DMA RUN, DATA POINTS WRITTEN = ',I8,
230. 1   'LAST SEQUENCE NO. = ',I10)
231.   WRITE(10UT,1092) NRECRD
232.   WRITE(10UT,1091) 10DBAL
233.   OUTPUT 'ALL DONE'
234.   CALL EXIT
235. C
236. C   FORMATS
237. C
238. 98 FORMAT(1X,32A4)
239. 1001 FORMAT(11,14,3I2,I4,2F9.4,F7.2,I3,F6.2,I5,2F6.1,F4.1,

```

```
240.      *      2I2,F6.1,I1,I2,35A1,1X,I1,2I3,I2)
241. 1091 FORMAT(3X,I5,1X,'BODDBALL RECORDS WRITTEN')
242. 1092 FORMAT(3X,I5,1X,'RECORDS READ')
243.      END
```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	00AC9 V	1	ALAT	R	SCALR	00AEC V	1	ALONG	R	SCALR	00AEF V	1
ALTD	SPR0G	EXTERN	00AF7 V	1	B	R	SCALR	00AF7 V	1	BG	R	SCALR	00AE6 V	1
BUFFIN	SPR0G	EXTERN	00AF7 V	1	BUFFOUT	R	SPR0G	00AF7 V	1	DEE	R	SCALR	00AB5 V	1
DEGRA	R	SCALR	00AC4 V	1	DERIN	R	SCALR	00AF2 V	1	DLAT	R	SCALR	00ACA V	1
DLONG	R	SCALR	00ACB V	1	EIGHT	R	SCALR	00AF2 V	1	ELEV	R	SCALR	00AE3 V	1
EXIT	R	SCALR	00ACB V	1	FA	R	SCALR	00AF5 V	1	FIVE	R	SCALR	00AB0 V	1
EXIT	R	SPR0G	EXTERN	1	FBR	R	SCALR	00AF5 V	1	G0BS	D	SCALR	00AAA V	2
FL0AT	R	SPR0G	INTRIN	1	IBUF	R	SCALR	00AF5 V	35	IAKEY	R	SCALR	00ACD V	1
HEIGT	R	SCALR	00ACC V	1	IBR	R	SCALR	00AF5 V	1	IBUFIN	R	SCALR	00023 V	1050
IBASE	R	SCALR	00AC4 V	1	ICHECK	R	SPR0G	EXTERN	1	ICNT	R	SCALR	00AC0 V	1
IBUF8T	R	SCALR	00AC4 V	1600	IDEP	R	SCALR	00AF4 V	1	IDIF	R	SCALR	00AD6 V	1
IDEIF	R	SCALR	00ACE V	1	IELEV	R	SCALR	00AF0 V	1	IELU	R	SCALR	00AF1 V	1
IELC	R	SCALR	00ADA V	1	IFBC	R	SCALR	00ADE V	1	IFFC	R	SCALR	00ADD V	1
IBD	R	SCALR	00ACE V	1	IGBC	R	SCALR	00AE9 V	1	IBUT	R	SCALR	00AFE V	1
IBC	R	SCALR	00AD5 V	1	IN	R	SCALR	00ABC V	1	IBDBAL	R	SCALR	00AB8 V	1
IMEY	R	SCALR	00AD5 V	1	IBUTSM	R	SCALR	00AC3 V	1	IREC2	R	SCALR	00ABE V	1
IREGC	R	SCALR	00ABD V	1	ISLG	R	SCALR	00AF6 V	1	ISL	R	SPR0G	INTRIN	1
ISLAT	R	SCALR	00ADC V	1	ISL	R	SCALR	00AF6 V	1	IS9RC	R	SCALR	00AD1 V	1
ITAPE	R	SCALR	00AE4 V	1	ISL	R	SCALR	00AF6 V	1	IZ	R	ARRAY	00AD7 V	9
JTAP	R	SCALR	00AB7 V	1	ISL	R	SCALR	00AF6 V	35	JNEG	R	SCALR	00AB6 V	1
JTAP	R	SCALR	00AB7 V	1	ISL	R	SCALR	00AF6 V	1	KGDA	R	SCALR	00AD2 V	1
KGDA0	R	SCALR	00AB8 V	1	ISL	R	SCALR	00AF6 V	1	KGHM9	R	SCALR	00AF8 V	1
KGH8	R	SCALR	00AD7 V	1	ISL	R	SCALR	00AF6 V	1	KGYR	R	SCALR	00AD4 V	1
KGYR8	R	SCALR	00AD3 V	1	ISL	R	SCALR	00AF6 V	1	KK	R	SCALR	00AC5 V	1
K0	R	SCALR	00AD9 V	1	ISL	R	SCALR	00AF6 V	1	K977	R	SCALR	00AFF V	1
LAT	R	SCALR	00AC7 V	1	ISL	R	SCALR	00AF6 V	1	L0NG	R	SCALR	00AEE V	1
LTAPE	R	SCALR	00AB9 V	1	ISL	R	SCALR	00AF6 V	1	ND	R	SCALR	00AE8 V	1
NI	R	SCALR	00AE1 V	1	ISL	R	SCALR	00AF6 V	1	NINCHK	R	SCALR	00AE2 V	1
NINE	R	SCALR	00AB4 V	1	ISL	R	SCALR	00AF6 V	1	NREC	R	SCALR	00AC8 V	1
NRECRD	R	SCALR	00ABA V	1	ISL	R	SCALR	00AF6 V	1	8BGD	R	SPR0G	EXTERN	1
8BSG	R	SCALR	00AB0 V	1	ISL	R	SCALR	00AF6 V	1	PLAT	R	SCALR	00AFA V	1
PL0NG	R	SCALR	00AFC V	1	ISL	R	SCALR	00AF6 V	1	SEVEN	R	SCALR	00AB2 V	1
SIX	R	SCALR	00AB1 V	1	ISL	R	SCALR	00AF6 V	1	TC	R	SCALR	00AE7 V	1
THREE	R	SCALR	00AB1 V	1	ISL	R	SCALR	00AF6 V	1	UNPKBY	R	SPR0G	EXTERN	1

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
8	000E1	10	00119	15	00124	20	00132	30	0013E	30	0013E	40	0014C	300	00273
50	00158	90	00158	95	00162	98	00345	99	00313	340	00255	350	002D1	510	00205
305	0029C	310	0029F	320	002AC	330	002B8	340	00255	507	001B1	550	00273	960	00313
410	001BF	420	001D3	500	00187	505	000F2	530	00271	950	0030A	5215	00257		
520	0021D	521	0025A	524	0025F	528	00268	1092	0035E						
910	002DA	920	002E7	930	002F3	940	002FE								
999	002DA	1001	00349	1090	0031E	1091	0035E								
5555	00253														

LOCAL VARIABLES (2819 WORDS):

00000 IA	00023 IBUFIN	00A7D IZ	00A86 IW	00AAA G0BS
00AAC 6NE	00AAC THREE	00AAF FBUR	00AB0 FIVE	00AB1 SIX
00AB2 SEVEN	00AB4 NINE	00AB5 DEE	00AB6 JNEG	00AB7 ITAPE
00AB9 JTAP	00ABA NRECRD	00AB8 I9DBAL	00ABC IN	00ABD IOUT
00ABE IREC2	00AC0 ICNT	00AC1 NIN	00AC2 NINUT	00AC3 IOUT4

00AC4 DEGRA	00AC5 KK	00AC6 KI	00AC7 K8	00AC8 NREC	00AC9 A
00ACA CLAT	00ACB DLNG	00ACC HEIGHT	00ACD IKEY	00ACE IDEIF	00ACF IE8D
00ADO NSEC	00AD1 ISRC	00AD2 KGDA	00AD3 KGM8	00AD4 KGYR	00AD5 KGHM
00AD6 IDIF	00AD7 KGDA8	00AD8 KGM88	00AD9 KGYR8	00ADA IELC	00ADB IGC
00ADC IREGC	00ADD IFPC	00ADE IFBC	00ADF RFA	00AE0 IKEY	00AE1 NI
00AEC NINCHK	00AE3 ELEV	00AE4 IDEP	00AE5 FA	00AE6 BG	00AE7 TC
00AEB ND	00AE9 IGE8C	00AEA ISLAT	00AEB LAT	00AEC ALAT	00AED ISLG
00AEE LANG	00AEF ALNG	00AF0 IELEV	00AF1 IELU	00AF2 DEPTN	00AF3 SOURCE
00AF4 IBASE	00AF5 IBR	00AF6 ISEG	00AF7 B	00AF8 KGHM8	00AF9 J
00AFA PLAT	00AF8 LKEY	00AFC PLNG	00AFD LGKEY	00AFE IIBUT	00AFF K977
00B00 8BSG	00B01 JKEY	00B02 JWD8			

BLANK COMP-8N (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

FL0AT ISL

EXTERNAL SUBPROGRAMS REQUIRED:

ALTD	BUFFIN	BUFF8UT	ICHECK	UNPKBY	F:101
F:102	F:103	F:104	F:106	M:D8	M:8C
9BCDREAD	9BCDWRIT	9DCODE	9ENDFILE	9INITIAL	9I8DATA
9I8LLSA	9IT8R	9PRINT	9ST8P		
		EXIT			
		F:105			
		9ENCODE			
		9RT8I			

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	882	DEC	HEX
CONSTANTS:	17	WORDS	WORDS
LOCAL VARIABLES:	2819	----	----
TEMPS:	1	00372	00011
		00B03	00001
		----	----
TOTAL PROGRAM:	3719	00E87	

```

1.  C      PROGRAM DMAP
2.  C      VERSION OF 8 DEC 1975 TO CHECK HEMISPHERES
3.  C      VERSION 27 OCT 1975
4.  C      VERSION AUGUST 1968
5.  C      IMPLICIT REAL*8(A-H,O-Z)
6.  C      DIMENSION XC(4),YC(4),                SC(4),CC(4)
7.  C      DIMENSION EDEG(8),BDIF(8)
8.  C      DIMENSION LAT(4),LONG(4)
9.  C      OUTPUT IDMAP = VERSION 8 DEC 1975
10. C      CONVERTS DIGITIZED POSITION IN INCHES
11. C      TO LATITUDE AND LONGITUDE
12. C      NO COMMON REQUIRED
13. C      ISW CHOICE DATA CARD GOES BEFORE ITAPE ETC CARD
14. C      USES SUB CALSC(ARG), FUNC PARTM(ARG), FUNC ISW(N),
15. C      SUB RTODM(ARG),FUNC DMTOR(ARG)
16. C
17. C      SSW(2) UP TO LIST INTERMEDIATE VALUES
18. C      SSW(3) UP TO LIST DATE AND SMIN FOR EACH DATA POINT
19. C      SSW(5) UP TO OUTPUT SMIN ONLY IF GREATER THAN EPSIL
20. C
21. C      NYR=0 WILL TERMINATE PROGRAM
22. C      START INITIALIZATION FOR SIGMA 7
23. C      IIN=105
24. C      IIBUT=108
25. C      INIT = ISW (-2)
26. C      END INITIALIZATION FOR SIGMA 7
27. C      DEGRA=1.745329E-2
28. C      IHENW=IHEMS=0
29. C      ITYPE=0
30. C      IGAL=0
31. C      N60=0
32. C      MAP=0
33. C      ITAPE = URN FOR DATA INPUT
34. C      JTAPE = URN FOR DATA OUTPUT
35. C      FFAC = FACTOR (0.1 TO 1.00) USED IN ITERATION FOR
36. C      ESTIMATED LATITUDE TO CONVERGE ON TRUE
37. C      LATITUDE.
38. C      EPSIL = TOLERANCE (IN MERIDIONAL PARTS) BY WHICH
39. C      ESTIMATED LATITUDE MUST MATCH MERIDIONAL
40. C      PARTS FOR TRUE LATITUDE.
41. C      READ(IIN,6)ITAPE,JTAPE,FFAC,EPSIL
42. C      6  FORMAT(2I5,F5.2,F5.2)
43. C      READ IN DATA
44. C      8  READ(ITAPE,15)ICODE,XP,YP,NDA,NMB,NYR,NHM
45. C      15  FORMAT(I1,1X,F5.3,1X,F5.3,3I3,15)
46. C      CHANGING SIGN FOR WESTERN OR SOUTHERN HEMISPHERE
47. C      IF(IHENW .GT.0) XP=-1.0*XP
48. C      IF(IHEMS .GT.0) YP=-1.0*YP
49. C      IF(ICODE=9)50,20,50
50. C      SETTING MAP COORDINATE AND SCALE
51. C      20  XC(1)=XP,YC(1)=YP,LAT(1)=NMB,LONG(1)=NYR
52. C      CHANGING SIGN FOR WESTERN OR SOUTHERN HEMISPHERE
53. C      IF(LONG(1).LT.0) XC(1)=-1.0*XC(1),IHENW=5
54. C      IF(LAT(1).LT.0) YC(1)=-1.0*YC(1),IHEMS=5
55. C      DO 25 J=2,4
56. C      READ(ITAPE,15)ICODE,XC(J),YC(J),N1,N2,LAT(J),LONG(J)
57. C      CHANGING SIGN FOR WESTERN OR SOUTHERN HEMISPHERE
58. C      IF(LONG(J).LT.0) XC(J)=-1.0*XC(J)

```

```

60.      25 CONTINUE
61.      MAP=MAP+1
62.      C DETERMINING AVERAGE SIN AND COS OF ANGLE OF TILT OF MAP
63.          CALL CALSC(XC(1),YC(1),XC(2),YC(2),SC(1),CC(1))
64.          CALL CALSC(YC(2),XC(3),YC(3),XC(2),SC(2),CC(2))
65.          CALL CALSC(XC(4),YC(4),XC(3),YC(3),SC(3),CC(3))
66.          CALL CALSC(YC(1),XC(4),YC(4),XC(1),SC(4),CC(4))
67.          AS = (SC(1)+SC(2)+SC(3)+SC(4))/4.0
68.          AC = (CC(1)+CC(2)+CC(3)+CC(4))/4.0
69.      C LIST CALCULATED SIN AND COS OF ANGLE OF TILT OF MAP
70.          WRITE(IIOUT,24)MAP,AS,AC
71.      24 FORMAT('MAP = ',I4,' SIN A = ',F8.6,' COS A = 'F8.6)
72.          IF(ISW(2))26,30,26
73.      26 WRITE(IIOUT,27)SC(1),SC(2),SC(3),SC(4)
74.          WRITE(IIOUT,27)CC(1),CC(2),CC(3),CC(4)
75.      27 FORMAT(4F10.6)
76.      C ROTATION OF AXES TO CALCULATE MAP SCALE
77.      30 XT=XC(2)-XC(1)
78.          YT=YC(4)-YC(1)
79.          YTT=DABS(YC(2)-YC(1))
80.          XTT=DABS(XC(4)-XC(1))
81.          X3=DSGRT((XT**2)+(YTT**2))
82.          Y3=DSGRT((YT**2)+(XTT**2))
83.          A=LONG(1)
84.          B=LONG(3)
85.          SINCH=DABS(X3/(B-A))
86.          SMP=SINCH/60.0
87.          AM=0.0
88.          RLEFT=DMT0R(LONG(1),AM)
89.          RBOT =DMT0R(LAT(1),AM)
90.          RRIGT=DMT0R(LONG(3),AM)
91.          RTOP =DMT0R(LAT(3),AM)
92.          FLEFT=LONG(1)
93.          FBOT=LAT(1)
94.          BOTMP=PARTM(RBOT)
95.          TOPMP=PARTM(RTOP)
96.          IF(RBOT)33,35,35
97.      33 BOTMP=-BOTMP
98.      35 IF(RTOP)36,40,40
99.      36 TOPMP=-TOPMP
100.     40 DLDEG=LAT(3)-LAT(1)
101.         AVMP=(TOPMP-BOTMP)/DLDEG
102.     C FINISHED CALCULATING SCALE AND PARAMETERS FROM COORDINATE POINTS
103.         GO TO 8
104.     C MAIN CALCULATION PORTION FOR MAP DATA POINTS
105.     50 IF(NYR)52,100,52
106.     C ROTATING DATA POINT
107.     52 XP=XP-XC(1)
108.         YP=YP-YC(1)
109.         X = (XP*AC)+(YP*AS)
110.         Y = -1.0*(XP*AS)+(YP*AC)
111.     C DETERMINING LONGITUDE OF DATA POINT
112.         DEGLG=FLEFT+(X/SINCH)
113.         ILONG=DEGLG
114.         A=ILONG
115.         RL0M=(DEGLG-A)*60.0
116.     C
117.     C DETERMINING LATITUDE OF DATA POINT
118.     C
119.         YY=Y

```

```

120.      PM=B0TMP+(YY/SMP)
121.      KCNT=0
122.  C  MAKING FIRST ESTIMATE OF LATITUDE OF DATA POINT
123.      DEGE=FB0T+((PM-B0TMP)/AVMP)
124.      60  RDEGE=DEGE*DEGRA
125.      CALMP=PARTM(RDEGE)
126.      KCNT=KCNT+1
127.  C  FINDING DIFFERENCE BETWEEN ACTUAL AND MERIDONAL PARTS FOR
128.  C      ESTIMATED LATITUDE
129.      DIFMP=PM-CALMP
130.      IF(ISW(2))61,65,61
131.      61  WRITE(IIBUT,62)KCNT,PM,DIFMP,X3,Y3,SINCH,
132.      1  SMP,AVMP,X,Y,YY,B0TMP,T0PMP
133.      62  FORMAT(14,6F10.2,6F10.2)
134.      65  ADIF=DABS(DIFMP)
135.      EDEG(KCNT)=RDEGE
136.      BDIF(KCNT)=ADIF
137.  C  EPSIL IS MERIDONAL PARTS FOR 0.1 MINUTE OF ARC
138.      IF(KCNT=8)68,80,80
139.  C  MAKING NEW ESTIMATE OF LATITUDE FOR DATA POINT
140.      68  DEGE=DEGE+((DIFMP/AVMP)*FFAC)
141.      GO TO 60
142.  C  SELECTING MINIMUM ADIF, CALLED SMIN
143.      80  SMIN=BDIF(1)
144.      RDEGE=EDEG(1)
145.      DO 82 I=2,8
146.      IF(BDIF(I)-SMIN,81,82,82)
147.      81  SMIN=BDIF(I)
148.      RDEGE=EDEG(I)
149.      82  CONTINUE
150.      IF(ISW(3))83,184,83
151.      83  WRITE(IIBUT,183)NDA,NM0,NYR,NHM,SMIN
152.      183  FORMAT(3I2,15,F7.2)
153.      GO TO 84
154.      184  IF(SMIN-EPSIL)84,84,185
155.      185  IF(ISW(5))83,84,83
156.      84  CALL RT0DM(RDEGE,ILAT,RLATM)
157.  C  OUTPUT POSITION OF DATA POINT
158.      WRITE(JTAPE,85)NDA,NM0,NYR,NHM,ILAT,RLATM,
159.      1  ILONG,RL0M,ITYPE,IQAL,N60
160.      85  FORMAT(3I2,14,5X,13,1X,F7.3,14,1X,F7.3,3X,12,11,15X,11)
161.  C  85  FORMAT(3I2,14,9X,13,F6.2,14,F6.2,2X,12,11,16X,11)  OLD FORMAT
162.      GO TO 8
163.      100  WRITE(IIBUT,101)
164.      101  FORMAT('LAST POINT PROCESSED:')
165.      END FILE JTAPE
166.      REWIND JTAPE
167.      END

```

[illegible]

ICAL VARIABLES (182 WORDS):

00000	XC	00008	YC	00010	SC	00018	CC	00020	EDEG	00030	BDIF
00040	LAT	00044	LANG	00048	IN	00049	IBUT	0004A	INIT	0004C	MAG
0004E	IHEM	0004F	IHEMS	00050	ITYPE	00051	IGAL	00052	N60	00053	DEGRA
00054	ITAPE	00055	ITAPE	00056	FFAC	00058	EPSIL	0005A	IC8DE	0005C	XP
0006E	YP	00060	NDA	00061	M9	00062	NYR	00063	NHM	00064	J
00065	N1	00066	N2	00068	AS	0006A	AC	0006C	XT	0006E	YT
00070	YTT	00072	XIT	00074	X3	00076	Y3	00078	A	0007A	B
0007C	SINCH	0007E	SMP	0008C	AM	00082	RLEFT	00084	R80T	00086	RRIGT
0008B	RT8P	0008A	FLEPT	0008C	F80T	0008E	B80T	00090	T80P	00092	DLDEG
00094	AVMP	00096	X	00098	Y	0009A	DEGLG	0009C	IL8M	0009E	RL8M
000A0	YY	000A2	PM	000A4	KCNT	000A6	DEGE	000A8	RDEGE	000AA	CALMP
000AC	DIFM	000AE	ADIF	000B0	SNIN	000B2	I	000B3	ILAT	000BA	FLATM

ANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

DABS DSGRT

INTERNAL SUBPROGRAMS REQUIRED:

CALSC	DMTBR	ISW	PARTM	RT8DM	F:101	F:102	F:103
F:104	F:105	F:106	F:108	M:08	M:0C	9BCDREAD	9BCDWRT
9DSGRT	9DT8I	9ENDFILE	9ENDI8L	9INITIAL	9I8DATA	9IT8D	9PRINT
9REWIND	9ST8P						

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	544	00220
CONSTANTS:	14	0000E
CAL VARIABLES:	182	000B6
TEMPS:	2	00002
TOTAL PROGRAM:	742	002E6

```

1.  C  PROGRAM DM0D
2.  C  VERSION OF 6 OCT 1972, TO ALSO HANDLE CASE WHERE SECOND CARD
3.  C  TO THE LEFT OF THE ORIGIN POINT (FIRST CARD)
4.  C  VERSION OF 15 OCT 1971, CORRECTS FORMAT ERROR
5.  C  VERSION OF 6 APRIL 1971, OUTPUTS POLYGON NUMBER IN OUTPUT CARDS
6.  C  VERSION OF 23 MARCH 1971, ALLOWS FOR MAKING END CARDS
7.  C  VERSION OF 8 FEB 1971
8.  C  PROGRAM DM0C, FOR DIGITIZING POLYGONS FOR CRUSTAL MODELS
9.  C
10. C
11. C  SSW(1) = 0 FOR SECOND POINT TO RIGHT (+) OF ORIGIN
12. C  = 1 FOR SECOND POINT TO LEFT (-) OF ORIGIN
13. C
14. C
15. C  ON DIGITIZING TABLE, IC0DE IS SET IN LEFTMOST THUMBWHEEL SWITCH
16. C  POSITION ON MANUAL ENTRY SWITCHES
17. C  POLYGON NUMBERS ARE SET IN THE THREE PAIRS OF SWITCHES
18. C  TO THE RIGHT OF THE LEFTMOST SWITCH
19. C
20. C  INFLT DATA CARDS =
21. C  1  VALUES FOR XFAC & YFAC IN KM'S / INCH AND
22. C      KM VALUES OF ORIGIN OF MODEL (4F10.0)
23. C  2  X AND Y VALUES FROM DIGITIZING TABLE FOR ORIGIN
24. C  3  X AND Y VALUES FROM DIGITIZING TABLE FOR A POINT AT SAME Y
25. C      LEVEL AS ORIGIN
26. C  4  X AND Y VALUES FROM DIGITIZING TABLE FOR POLYGON CORNERS
27. C
28. C  SET IC0DE = 9 FOR INDICATING LAST CARD OF POLYGON FOR TALPLOT PR0G
29. C  SET IC0DE = 8 FOR X = -3000 KM
30. C  SET IC0DE = 7 FOR X = +3000 KM
31. C
32. C  IC0DE IS RESET BY PROGRAM TO ZERO
33. C
34. C  LAST DATA CARD SHOULD HAVE IC0DE = 99
35. C
36. C
37. C  OUTPUT : DM0D OF 6 OCT 1972
38. C  IIN= 105
39. C  II0UT= 108
40. C  JTAPE= 106
41. C  INIT = ISW(*2)
42. C  READ(IIN,12) XFAC,YFAC,X0RG,Y0RG
43. C  12  FORMAT(4F10.0)
44. C  READ(IIN,20)XA,YA,IA ,KF1,KP2,KP3
45. C  20  FORMAT(2F10.3,I5,3I4)
46. C  READ(IIN,20)XB,YB,IB ,KF1,KP2,KP3
47. C  CALL CALSC(XA,YA,XB,YB,AS,AC)
48. C  OUTPUT AS,AC
49. C  50  READ(IIN,20)XP,YP,IC0DE, KP1,KF2,KP3
50. C  IF(IC0DE-90)60 , 60 , 999
51. C  60  CONTINUE
52. C  XP= XP-XA
53. C  YP= YP-YA
54. C  X= (XP * AC) + (YP * AS)
55. C  Y= -1.0 *(XP*AS)+(YP*AC)
56. C  XKM= (X*XFAC) + X0RG
57. C  YKM= (Y*YFAC)+ Y0RG
58. C  YKM=-1.0 * YKM
59. C  IF(ISW(1))68,68,64

```

```

60.  C
61.  C      REVERSE SIGN OF X DISTANCE SINCE SECOND LEVELING POINT
62.  C      WAS TO THE LEFT OF ORIGIN
63.  C
64.      64  XKM=-XKM
65.      YKM=-YKM
66.      68  CONTINUE
67.      IF(ICODE=8)74,70,74
68.      7C  XKM=-3000.0
69.      ICODE=C
70.      GO TO 100
71.      74  IF(ICODE=7)90,76,90
72.      76  XKM=+3000.0
73.      ICODE=0
74.      GO TO 100
75.      9C  CONTINUE
76.      10C IF(KP1)110,110,105
77.      105 WRITE(,TAPE,22)XKM,YKM,ICODE,KP1
78.      11C IF(KP2)120,120,115
79.      115 WRITE(,TAPE,22)XKM,YKM,ICODE,KP2
80.      12C IF(KP3)130,130,125
81.      125 WRITE(,TAPE,22)XKM,YKM,ICODE,KP3
82.      13C CONTINUE
83.      22  FORMAT(2F10.2,I1,20X,I5)
84.      24  FORMAT(2F10.2,I3,3I10)
85.      WRITE(IIBLT,24) XKM,YKM,ICODE,KP1,KP2,KP3
86.      GO TO 50
87.      999  CALL EXIT
88.      END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
AC	R	SCALR	00012 V	1	AS	R	SCALR	00011 V	1	CALSC	I	SPR8G	EXTERNAL	1
EXIT	I	SPR8G	EXTERNAL	1	IA	I	SCALR	0000A V	1	IB	I	SCALR	00001 V	1
IC0DE	I	SCALR	00015 V	1	IIN	I	SCALR	00000 V	1	IIBUT	I	SCALR	00002 V	1
INIT	I	SCALR	00003 V	1	ISW	I	SPR8G	EXTERNAL	1	JTAPE	I	SCALR	00000 V	1
KP1	I	SCALR	00008 V	1	KP2	R	SCALR	0000C V	1	KP3	R	SCALR	0000E V	1
X	R	SCALR	00016 V	1	XA	R	SCALR	00008 V	1	XB	R	SCALR	00006 V	1
XFAC	R	SCALR	00004 V	1	XKM	R	SCALR	00017 V	1	Y	R	SCALR	00009 V	1
XP	R	SCALR	00013 V	1	YFAC	R	SCALR	00005 V	1	YKM	R	SCALR	00019 V	1
YB	R	SCALR	0000F V	1	YP	R	SCALR	00014 V	1					
Y8RG	R	SCALR	00007 V	1										

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
12	0001D	22	000B6	50	0004E
64	00082	70	00088	76	00093
100	00038	110	000A2	120	000AC
130	000B6				

## LOCAL VARIABLES (26 WORDS):

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
00000 IIN	00001 IIBUT	00002 JTAPE	00003 INIT	00004 XFAC	00005 YFAC
00006 X8RG	00007 Y8RG	00008 XA	00009 YA	0000A IA	0000B KP1
0000C KP2	0000D KP3	0000E XB	0000F YB	00010 IB	00011 AS
00012 AC	00013 XP	00014 YP	00015 IC0DE	00016 X	00017 Y
00018 XKM	00019 YKM				

## BLANK COMBON (0 WORDS)

## EXTERNAL SUBPROGRAMS REQUIRED:

CALSC	EXIT	ISW	F:101	F:102	F:103	F:104	F:105
F:106	F:108	M:DB	M:BC	9BCDREAD	9BCDWRIT	9ENDI8L	9INITIAL
9I8DATA	9PRINT	9STEP					

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
208	00000
5	00005
26	0001A
1	00001
240	000FC

TOTAL PROGRAM: 240

COMPILED 6 DEC 73

```

1.  CS   PROGRAM GFLD1
2.  C     VERSION OF 22 SEPT 1972, TO CORRECT PROGRAM NAME IN OUTPUT
3.  C     SOURCE OF OCT 19, 1970
4.  C
5.  C   PROGRAM GFLD, SPHERICAL HARMONIC CALCULATION IN A
6.  C     GIVEN REGION
7.  C
8.  C     INCLUDING READING OF C,S
9.  C     SSW(4) UP TO LIST DATA ON TTY DURING RUN
10. C
11. C   OUTPUTS REGIONAL FA VALUE IN FREE-AIR POSITION OF
12. C     SEAG1 FORMAT
13. C
14. C     USES SUBROUTINES ISW, FLD2
15. C
16. C
17. C
18. 999  IIN = 105
19.      IIBUT = 108
20. C
21. C   OUTPUT : GFLD1 RUN, VERSION OF 22 SEPT 1972
22. C     II = ISW(-2)
23. C     READ (IIN, 8) ITAP, JTAPE
24. C     FORMAT (2I5)
25. C     OUTPUT ITAPE, JTAPE
26. C     IREC=1
27. C     KGYR=22
28. C     KK=0
29. CS600 WRITE(IIBUT, 9)
30. CS 9   FORMAT('E ITOP IBOT ILEFT IRIGHT INC')
31. CS     READ(IIN, *) ITOP, IBOT, ILEFT, IRIGHT, INC
32. 600   READ(IIN, 9) ITOP, IBOT, ILEFT, IRIGHT, INC
33. 9     FORMAT (5I5)
34. C     OUTPUT ITOP, IBOT, ILEFT, IRIGHT, INC
35. C     CALL FLD2(KK, ITAPE, RLAT, RLONG, REG)
36. C     IAREA=0
37. C     ISURV=0
38. C     ISTA=0
39. C     ELEV=0.0
40. C     K977=0.0
41. C     GBBS=0.0
42. C     DEPTH=0.0
43. C     FA=999.0
44. C     BG=999.0
45. C     TC=99.0
46. C     BGCBM=999.0
47. C     DEGRA=1.745329E-2
48. CS     ANS=123B
49. CS     NEX=127B
50. CS     UN=116B
51. CS     US=123B
52. CS     UE=105B

```

```

53.  CS      JW=1278
54.  CS      CONTINUE
55.      399  DB 402 I=I8BT,IT8P,INC
56.      DB 402 J=ILEFT,IRIGT,INC
57.      DLAT=I
58.      DLBN=J
59.      RLAT=DLAT*DEGRA
60.      RLONG=DLBN*DEGRA
61.      KK=1
62.      CALL FLD2(KK,ITAPE,RLAT,RLONG,REG)
63.      306  IF(ISW(4))3306,3308,3306
64.      3306  WRITE(IIOUT,307)DLAT,DLBN,REG
65.      307  FORMAT('DLAT=',F7.2,' DLBN=',F7.2,' REG=',F8.3)
66.      3308  FA=REG
67.      KFA=FA*10.0
68.  CS 445  CALL ENDI8
69.  C  OUTPUT AT SEAG1 FORMAT
70.      IF( ISW(26).EQ.1 .OR. JTAPE.EQ.108) IREC=6 ; JTAPE=108
71.      WRITE(JTAPE,12)IREC,KGYR,RLAT,RLONG,KFA
72.      12  FORMAT(I1,4X,I2,7X,2F9.6,17X,I5,35X)
73.      402  CONTINUE
74.      WRITE(IIOUT,410)
75.      410  FORMAT('THIS RUN COMPLETED'//)
76.      ENDFILE JTAPE
77.      STOP
78.      END

```



HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	234	000EA
CONSTANTS:	7	00007
LOCAL VARIABLES:	33	00021
TEMPS:	2	00002
TOTAL PROGRAM:	276	00114

(NO MEMORY PROTECTION)

COMPILED 13 MAR 73

```

1.  C      PROGRAM GFLD2
2.  C      VERSION OF 4 JANUARY 1972
3.  C      VERSION OF 13 DEC. 1971, TO READ COEFFICIENTS FROM CARDS
4.  C      VERSION OF 26 NOV 19718 FOR READ AND WRITE OF GSUM FMT
5.  C
6.  C      PROGRAM GFLD, SPHERICAL HARMONIC CALCULATION IN A
7.  C      GIVEN REGION
8.  C
9.  C      INCLUDING READING OF C,S
10. C      SS*(4) UP TO LIST DATA ON TTY DURING RUN
11. C
12. C      OUTPUTS REGIONAL FREE-AIR VALUE IN REGIONAL POSITION OF GSUM
13. C
14. C      USES SUBROUTINES ISW, FLD2,GINBT
15. C
16. C
17. C
18. C      DIMENSION IA(35)
19. C      DIMENSION N8W(4)
20. C      CALL STAT
21. C      IIN = 105
22. C      IIBUT = 108
23. C      ITAPE=1
24. C      JTAPE=2
25. C      KTAPE=105
26. C      NOUT=0
27. C      PRINT DATE AND TIME OF JOB ON HEADING
28. C      CALL TODAY(N8W)
29. C      WRITE(IIBUT,13) N8W
30. C      13  FORMAT(1X,4A4)
31. C
32. C      OUTPUT : GFLD2 RUN, VERSION OF 13 DEC 1971
33. C      INIT=ISW(-2)
34. C      KK=0
35. C      CALL FLD2(KK,KTAPE,RLAT,RLONG,REG)
36. C      CALL      GINBT(ITAPE,JTAPE,KK,KGDA,KGM8,
37. C      1  KGYR,KGHM,IDIF,IS0RC,RLAT,RLONG,ELEV,K977,0BSG,
38. C      2  IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
39. C
40. C
41. C      READ INPUT DATA
42. C
43. C      10C  CONTINUE
44. C      KK=1
45. C      CALL      GINBT(ITAPE,JTAPE,KK,KGDA,KGM8,
46. C      1  KGYR,KGHM,IDIF,IS0RC,RLAT,RLONG,ELEV,K977,0BSG,
47. C      2  IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
48. C      IF(KK=9)120,540,120
49. C      12C  CONTINUE
50. C      CALL FLD2(KK,ITAPE,RLAT,RLONG,REG)
51. C      RFA=REG
52. C

```

```

53.  C   IREGC = 1 FOR SAO MODEL EARTH 1969 COEF'S FOR INTL GRAVITY FORMULA
54.  C
55.      IREGC=1
56.      KK=-2
57.      CALL      GINBT(ITAPE,JTAPE,KK,KGDA,KGM8,
58.      1      KGYR,KGHM,IDIF,ISORC,RLAT,RLONG,ELEV,K977,8BSG,
59.      2      IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
60.      NOUT=NOUT+1
61.      155  GO TO 100
62.      54C  WRITE(I1OUT,545,NOUT
63.      545  FORMAT('EOF FOUND ON INPUT TAPE ',I10)
64.      NOUT=0
65.      END FILE JTAPE
66.      STOP
67.      END

```



HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS	
	-----	-----	
GENERATED CODE:	176	C0080	(NO MEMORY PROTECTION)
CONSTANTS:	1	C0001	
LOCAL VARIABLES:	69	00045	
TEMPS:	0	C0000	
	-----	-----	
TOTAL PROGRAM:	246	C00F6	

```

1.  C   PROGRAM GRAFG2
2.  C   *****
3.  C   OUTPUT ' GRAFG2 RUN, VERSION 0F 11 AUG 75'
4.  C   *****
5.  C
6.  C   VERSION 11 AUG 75 TO ADD OUTPUT 0F NUMBER 0F RECORDS
7.  C   VERSION 11 APRIL 75, ADD TEST FOR SPACECRAFT ALTITUDE LUNAR DATA
8.  C   VERSION 9 APRIL 75, TO ADD SSW(3) FOR INPUT 0F LUNAR DATA
9.  C   VERSION 0F 19 JULY 1973, TO MOVE LABEL AWAY FROM JOB NO.
10. C   VERSION 0F 27 APRIL 1973, ADDING LABEL AND NOW DATE
11. C   PROGRAM GRAFG2, PLOTS ONE VARIABLE VS ANOTHER
12. C   ALSO HAS BEGINNING AND END DATE CHECK LOGIC
13. C
14. C   SSW(0) = 1 TO OUTPUT VALLES FOR TESTING
15. C   SSW(3) = 1 TO USE GETL FOR INPUT 0F LUNAR DATA
16. C   SSW(4) = 1 TO READ SVEC ALTITUDE BOUNDS FOR LUNAR DATA,
17. C   (AFTER READING AREA BOUNDS), AND TO PROCESS ONLY
18. C   DATA WITHIN THESE BOUNDS
19. C   SSW(7) = 1 TO INPUT NEW AREA BOUNDS FOR NEXT PLOT
20. C   SSW(8) = 1 TO SUPPRESS PLOTTING GRID
21. C   SSW(10) = 1 TO START A NEW GRAPH
22. C   SSW(12) = 1 TO LIST DATA IDENTIFICATION
23. C   SSW(13) = 1 TO ANNOTATE PLOT POINT WITH DATA
24. C
25. C   TO STOP THE RUN AFTER LAST DATE BLOCK HAS BEEN PROCESSED,
26. C   SET ISTDA 0F NEXT STARD/END DATE CARD = 99
27. C
28. C
29. C   DIMENSION IBUF(1000)
30. C   DIMENSION NOW(4)
31. C   DIMENSION LABEL(20)
32. C   INREC = 0
33. C   IGRAC = 0
34. C   IAREC = 0
35. C   IPREC = 0
36. C
37. C   PROGRAM TESTS FOR AREA AND GRAPH LIMITS
38. C
39. C   USES GRIDG, SPOT, ISW, STAT, GETG, GETL
40. C   ENDIO, EVIL, SHTV, AND CALCOMP ROUTINES
41. C
42. C
43. C   IIN = 105
44. C   IIOUT = 108
45. C   READ IN INFORMATION FOR PLOT LABEL
46. C   INPT=IIN
47. 5021 READ(INPT,5022) LABEL
48. 5022 FORMAT(20A4)
49. C   WRITE(IIOUT,5023) LABEL
50. 5023 FORMAT(1X,20A4)
51. C   INITIALIZE PSEUDO-SWITCHES AND PLOTTER ROUTINE
52. C   CALL PLOTS (IBUF, =1000)
53. C   INIT = ISW(=2)
54. C   PLNUM=1.0
55. C   PRINT DATE AND TIME 0F JOB ON HEADING
56. C   CALL TODAY(NOW)
57. C   WRITE(IIOUT,11) NOW
58. 11 FORMAT(1X,4A4)
59. C   CALL STAT

```

AD-A035 454

WHOI-77-2  
GRAVITY DATA PROCESSING PROGRAMS.(U)

FEB 77

CARL BOWIN

UNCLASSIFIED WOODS HOLE OCEANOGRAPHIC INSTITUTION, MA

3 OF 6  
AD-A  
035 454



```

60.      CALL SETSKP(IND)
61.      C
62.      C NX = PLT(NX) FOR X VARIABLE
63.      C NY = PLT(NY) FOR Y VARIABLE
64.      C NZ = PLT(NZ) FOR Z VARIABLE
65.      C NW = PLT(NW) FOR W VARIABLE
66.      C XFAC = ENGINEERING UNITS PER INCH ON PLOT FOR X DIRECTION
67.      C YFAC = ENGINEERING UNITS PER INCH ON PLOT FOR Y DIRECTION
68.      C ZFAC = SAME FOR Z DIRECTION
69.      C WFAC = SAME FOR W DIRECTION
70.      C ANGB = ANGLE FOR DATAW ANNOTATION
71.      C IDEC = PLOTTER CONTROL CODE FOR DECIMAL POINT IN DATAW ANOT
72.      C XINC = SPACING IN DECIMAL INCHES FOR ANOT IN X DIRECTION
73.      C YINC = SPACING IN DECIMAL INCHES FOR ANOT IN Y DIRECTION
74.      C
75.      READ (IIN, 2) ITAPE, NX, NY, NZ, NW, IDEC, KPT, KHT
76.      2      FORMAT (8I5)
77.      OUTPUT ITAPE, NX, NY, NZ, NW, IDEC, KPT, KHT
78.      READ (IIN, 3) XFAC, YFAC, ZFAC, WFAC, ANGB, XINC, YINC
79.      3      FORMAT (7F10.0)
80.      OUTPUT XFAC, YFAC, ZFAC, WFAC, ANGB, XINC, YINC
81.      ZZ=KPT
82.      ZHT=KPT*KHT
83.      HGT=0.07*ZHT
84.      XINC=XINC*ZZ
85.      YINC=YINC*ZZ
86.      XFAC=XFAC/ZZ
87.      YFAC=YFAC/ZZ
88.      ZFAC=ZFAC/ZZ
89.      WFAC=WFAC/ZZ
90.      READ (IIN, 4) TBP, BOT, DLEFT, RIGHT
91.      4      FORMAT (4F10.0)
92.      OUTPUT TBP, BOT, DLEFT, RIGHT
93.      TBPY=TBP/YFAC
94.      BOTY=BOT/YFAC
95.      DLEFX=DLEFT/XFAC
96.      RIGTX=RIGHT/XFAC
97.      LN=1
98.      6      READ (IIN, 5) IDEG, AMIN
99.      5      FORMAT (I5, F10.0)
100.      RADN=DMTBR(IDEG,AMIN)
101.      GO TO(110,120,130,140),LN
102.      110    RTBP=RADN
103.      LN=2
104.      GO TO 6
105.      120    RBOT=RADN
106.      LN=3
107.      GO TO 6
108.      130    RLEFT=RADN
109.      LN=4
110.      GO TO 6
111.      140    RRIGT=RADN
112.      402    CONTINUE
113.      C
114.      C READ SPACECRAFT ALTITUDE (SVEC) BOUNDS IF SSW(4) =1
115.      C
116.      IF(ISW(4).EQ.0) GO TO 407
117.      READ (IIN,403) BSVEC,TSVEC
118.      C
119.      C BSVEC IS LOWER ALTITUDE LIMIT, TSVEC IS UPPER

```

```

120. 403 FORMAT (2F10.3)
121. C SET NZ = THAT FOR SVEC
122. NZ = 1
123. 407 IFLAG = 0
124. READ(IIN,9)ISTDA,ISTMB,ISTYR,ISTHM,IENDA,IENMB,IENYR,IENHM,ISKP
125. 9 FORMAT( 3I2,I4,5X,3I2,I4,5X,I5)
126. IF(ISTDA-99)409,305,305
127. 409 WRITE(IIOUT,6365)ISTDA,ISTMB,ISTYR,ISTHM,IENDA,IENMB,IENYR,
128. 1 IENHM,ISKP
129. 6365 FORMAT('GRAFG2: START DATE ',3I2,I4,', END DATE ',3I2,I4,', ISKP=
130. 1',I4)
131. IF(ISKP.EG.C) GO TO 8
132. CALL SKPREC(ITAPE,ISKP)
133. GO TO(999,8,999,999,999,999) IND
134. 8 CONTINUE
135. 1C WRITE(IIOUT,7)PLNUM
136. 7 FORMAT('START PLOT NUMBER = ',F4.0)
137. CS CALL WHERE(XORG,YORG)
138. CALL WHERE(XORG,YORG,RFACT)
139. CALL PLOT(XORG,YORG,-3)
140. C ANNOTATING PLOT NUMBER
141. CALL NUMBER(0.0,0.0,0.0,14,PLNUM,0.0,-1)
142. C
143. C ANNOTATE PLOT WITH LABEL AND DATE
144. C
145. CALL SYMBOL(-2.0,0.0,0.0,14,LABEL,90.,80)
146. 505C CALL SYMBOL(-1.0,0.0,0.0,7,NBW,90.,16)
147. 505S CALL PLOT(0.0,0.0,-3)
148. C
149. C MOVING PEN TO ORIGIN OF PLOT
150. YT=-(BOT/YFAC)
151. IF(YT)205,210,210
152. 205 YT=0.0
153. 21C XT=-(DLEFT/XFAC)
154. IF(XT)215,220,220
155. 215 XT=0.0
156. 22C XT=XT+1.0
157. YT=YT+1.0
158. CALL PLOT(XT,YT,3)
159. CALL GRIDG(ZZ,HGT,XFAC,YFAC,TOP,BOT,DLEFT,RIGHT)
160. PLNUM = PLNUM + 1.0
161. 15 CONTINUE
162. C START NEW PLOT
163. 22 IF (ISW(10))24,26,24
164. 24 CALL PLOT(8.0,0.0,-3)
165. GO TO 10
166. 26 CONTINUE
167. 30C IF(ISW(3).EG.C) GO TO 302
168. CALL GETL (ITAPE, NX, NY, NZ, NW, DATA, DATAY, DATAZ, DATAW, RLAT,
169. 1 RLONG, JDA, JMB, JYR, JHM, IE0D)
170. GO TO 304
171. 302 CALL GETG (ITAPE, NX, NY, NZ, NW, DATA, DATAY, DATAZ, DATAW, RLAT,
172. 1 RLONG, JDA, JMB, JYR, JHM, IE0D)
173. 304 IF(IE0D)350,350,305
174. C EOF OR EOT ENCOUNTERED DURING READ --- QUIT
175. 305 CALL PLOT (XX, YY, 999)
176. WRITE (IIOUT,1020) INREC,IGREC,IAREC,IPREC
177. 102C FORMAT('NUMBER RECORDS INPUT = ',I,/,
178. 1 'NUMBER RECORDS OUT OF GRAPH BOUNDS = ',I,/,
179. 2 'NUMBER RECORDS OUT OF AREA BOUNDS = ',I,/,

```

```

180.      3 NUMBER RECORDS PLOTTED = 1,1)
181.      CALL EXIT
182.  C   CHECKING FOR BEGINNING DATE
183.      350 CONTINUE
184.      INREC = INREC + 1
185.      KGDA = JDA
186.      KGM0 = JMO
187.      KGYR = JYR
188.      KGHM = JHM
189.      186 CONTINUE
190.  C   IFLAG IS A FLAG TO ALLOW SKIPPING THE FIRST CALL TO FIND
191.  C   IF WE HAVE ALREADY FOUND THE STARTING DATE
192.  C
193.      IF (IFLAG.NE.0) GO TO 182
194.      CALL FIND(ISTDA,ISTMO,ISTYR,ISTHM,KGDA,KGM0,KGYR,KGHM,INDK)
195.      IF (INDK.EQ.-1) GO TO 300
196.      IFLAG=1
197.      182 CONTINUE
198.      IF (IENYR.EQ.0) GO TO 851
199.      CALL FIND(IENDA,IENMO,IENYR,IENHM,KGDA,KGM0,KGYR,KGHM,INDK)
200.      IF (INDK.EQ.1) GO TO 995
201.      851 CONTINUE
202.      852 CONTINUE
203.  C   CHECKING IF DATA WITHIN CHART BOUNDRIES
204.      85 IF (ISW(0))102,104,102
205.      102 WRITE(IIOUT,103)RLAT,RLONG,RTOP,RBOT,RLEFT,RRIGT
206.      103 FORMAT(6E12.5)
207.      104 IF (RTOP=RLAT) 100,100,86
208.      86 IF (RLAT=RBOT) 100,88,88
209.      88 IF (RLONG=RLEFT) 100,90,90
210.      90 IF (RRIGT=RLONG) 100,100,92
211.  C   DATA WITHIN BOUNDS
212.      92 GO TO 35
213.  C   DATA OUT OF AREA BOUNDS
214.      100 IAREC = IAREC + 1 ; GO TO 890
215.      35 CONTINUE
216.  C
217.  C   CHECKING IF LUNAR DATA IS WITHIN ALTITUDE BOUNDS
218.  C
219.      IF (ISW(4).EQ.0) GO TO 45
220.      IF ((DATAZ=BSVEC).LE.0) GO TO 890
221.      IF ((TSVEC=DATAZ).LE.0) GO TO 890
222.      45 XX = DATAZ/XFAC
223.      YY = DATAZ/YFAC
224.  C   CHECKING IF DATA IS IN GRAPH BOUNDS
225.      IND=0
226.      IF (TOPY=YY)50,50,52
227.      50 YY=TOPY
228.      IND=1
229.      52 IF (YY=BOTY)54,54,60
230.      54 YY=BOTY
231.      IND=1
232.      60 IF (XX=DLEFX)62,62,66
233.      62 XX=DLEFX
234.      IND=1
235.      66 IF (RIGTX=XX)68,68,69
236.      68 XX=RIGTX
237.      IND=1
238.  C   IF IND = 1 DATA POINT IS OUT OF GRAPH BOUNDS
239.      69 IF (IND.EQ.1) IGREC = IGREC + 1 ; GO TO 890

```

```

240.      8C  CALL PLOT(XX,YY,3)
241.      CALL SPOT(XX,YY)
242.      IPREC = IPREC + 1
243.      IF(ISH(13))82,890,82
244.      82  XT=XX+XINC
245.      YT=YY+YINC
246.      CALL NUMBER (XT, YT, HGT, DATAW, ANGB, IDEC)
247.      CALL PLOT(XX,YY,3)
248.      89C  CONTINUE
249.      90C  IF(IENYR.EQ.0) GO TO 951
250.      CALL FIND(IENDA,IENMO,IENYR,IENHM,KGDA,KGMO,KGYR,KGHM,INDK)
251.      IF(INDK.EQ.0) GO TO 940
252.      GO TO 150
253.      94C  XT=RIGHTX+5.0
254.      YT=BOTY
255.      CALL PLOT(XT,YT,3)
256.      IF(ISH(7))402,402,6
257.      951  GO TO 150
258.      995  WRITE(IIOUT,996) KGDA,KGMO,KGYR,KGHM
259.      996  FORMAT('END DATE PASSED',2X,3I2,14)
260.      CALL PLOT(XX,YY,999)
261.      CALL EXIT
262.      999  WRITE(IIOUT,998) IND
263.      998  FORMAT('ERROR IN SKPREC,IND=',1,13)
264.      CALL PLOT(XX,YY,999)
265.      CALL EXIT
266.      15C  GO TO 15
267.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
AMIN	R	SCALR	00426 V	1	ANBR	R	SCALR	00416 V	1	BOT	R	SCALR	0041D V	1	BOT	R	SCALR	0041D V	1
887Y	R	SCALR	00421 V	1	BSVEC	R	SCALR	0042C V	1	DATAM	R	SCALR	00410 V	1	DATAM	R	SCALR	00410 V	1
CLEFT	R	SCALR	0043D V	1	DATY	R	SCALR	0043E V	1	DATZ	R	SCALR	0043F V	1	DATZ	R	SCALR	0043F V	1
EXIT	R	SCALR	0041E V	1	DLEFX	R	SCALR	00422 V	1	DMDR	R	SPRGG	EXTERNAL	1	DMDR	R	SPRGG	EXTERNAL	1
GETL	R	SPRGG	EXTERNAL	1	FIND	R	SPRGG	EXTERNAL	1	GETG	R	SPRGG	EXTERNAL	1	GETG	R	SPRGG	EXTERNAL	1
IAREC	I	SCALR	00402 V	1	GRIG	R	SPRGG	EXTERNAL	1	HGT	R	SCALR	0041B V	1	HGT	R	SCALR	0041B V	1
IDEG	I	SCALR	00425 V	1	IBUF	R	SCALR	00400 V	1000	IDEC	R	SCALR	0040F V	1	IDEC	R	SCALR	0040F V	1
IENM0	I	SCALR	00434 V	1	LEACA	R	SCALR	00433 V	1	IENM	R	SCALR	00436 V	1	IENM	R	SCALR	00436 V	1
IFLAG	I	SCALR	0042E V	1	LEAYR	R	SCALR	00435 V	1	IEDD	R	SCALR	00407 V	1	IEDD	R	SCALR	00407 V	1
IIBUT	I	SCALR	00405 V	1	IGREC	R	SCALR	004C1 V	1	IIN	R	SCALR	00404 V	1	IIN	R	SCALR	00404 V	1
INIT	I	SCALR	00407 V	1	IND	R	SCALR	004C9 V	1	INDK	R	SCALR	0040E V	1	INDK	R	SCALR	0040E V	1
IPREC	I	SCALR	00403 V	1	INPT	R	SCALR	004C6 V	1	INREC	R	SCALR	00400 V	1	INREC	R	SCALR	00400 V	1
ISTHM	I	SCALR	00432 V	1	ISKP	R	SCALR	00437 V	1	ISTDA	R	SCALR	0042F V	1	ISTDA	R	SCALR	0042F V	1
ISM	I	SCALR	EXTERNAL	1	ISTMB	R	SCALR	00430 V	1	ISTVR	R	SCALR	00431 V	1	ISTVR	R	SCALR	00431 V	1
JHM	I	SCALR	00446 V	1	JTAP	R	SCALR	0040A V	1	JDA	R	SCALR	00403 V	1	JDA	R	SCALR	00403 V	1
KGDA	I	SCALR	0044A V	1	JMO	R	SCALR	00444 V	1	JYR	R	SCALR	00405 V	1	JYR	R	SCALR	00405 V	1
KGYR	I	SCALR	0044C V	1	KGM	R	SCALR	0044D V	1	KGM0	R	SCALR	0040B V	1	KGM0	R	SCALR	0040B V	1
LABEL	I	ARRAY	003EC V	20	KHT	R	SCALR	00411 V	1	KPT	R	SCALR	00410 V	1	KPT	R	SCALR	00410 V	1
NUMBER	I	SPRGG	EXTERNAL	1	LN	R	SCALR	00424 V	1	NGW	R	ARRAY	003E8 V	4	NGW	R	ARRAY	003E8 V	4
NY	I	SCALR	0040C V	1	NW	R	SCALR	004CE V	1	NX	R	SCALR	00408 V	1	NX	R	SCALR	00408 V	1
PLBT	R	SPRGG	EXTERNAL	1	NZ	R	SCALR	004CD V	1	PLNUM	R	SCALR	00427 V	1	PLNUM	R	SCALR	00427 V	1
REBT	R	SCALR	00429 V	1	PLBTS	R	SPRGG	EXTERNAL	1	RADN	R	SCALR	0042F V	1	RADN	R	SCALR	0042F V	1
RIGTX	R	SCALR	00423 V	1	RFAC	R	SCALR	0043A V	1	RIGT	R	SCALR	0041F V	1	RIGT	R	SCALR	0041F V	1
RLNG	R	SCALR	00442 V	1	RHAT	R	SCALR	00441 V	1	RLEFT	R	SCALR	0042A V	1	RLEFT	R	SCALR	0042A V	1
SETSKP	R	SPRGG	EXTERNAL	1	RHGT	R	SCALR	0042B V	1	RTPP	R	SCALR	00428 V	1	RTPP	R	SCALR	00428 V	1
STAT	R	SPRGG	EXTERNAL	1	SPREC	R	SPRGG	EXTERNAL	1	SPOT	R	SPRGG	EXTERNAL	1	SPOT	R	SPRGG	EXTERNAL	1
TBP	R	SCALR	0041C V	1	SYBBL	R	SCALR	00420 V	1	TDAY	R	SCALR	0042D V	1	TDAY	R	SCALR	0042D V	1
WFAC	R	SCALR	00415 V	1	TOPY	R	SCALR	00420 V	1	TSVEC	R	SCALR	0042D V	1	TSVEC	R	SCALR	0042D V	1
XINC	R	SCALR	00417 V	1	WHERE	R	SPRGG	EXTERNAL	1	XFAC	R	SCALR	00412 V	1	XFAC	R	SCALR	00412 V	1
XX	R	SCALR	00448 V	1	XORG	R	SCALR	00438 V	1	XT	R	SCALR	0043C V	1	XT	R	SCALR	0043C V	1
YORG	R	SCALR	00439 V	1	YFAC	R	SCALR	00413 V	1	YINC	R	SCALR	00418 V	1	YINC	R	SCALR	00418 V	1
ZFAC	R	SCALR	00414 V	1	YTH	R	SCALR	0043B V	1	YY	R	SCALR	00449 V	1	YY	R	SCALR	00449 V	1
					ZHT	R	SCALR	0041A V	1	ZZ	R	SCALR	00419 V	1	ZZ	R	SCALR	00419 V	1

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
2	CC057	7	000FB	5	CC0D3	11	CC043	5	CC0D3	11	CC043	5	CC0D3	11	CC043	5	CC0D3	11	CC043
8	CC17C	15	001D9	11	CC07C	45	CC09C	11	CC07C	45	CC09C	11	CC07C	45	CC09C	11	CC07C	45	CC09C
24	CC1DE	50	002A7	45	CC091	66	CC0B5	66	CC091	66	CC0B5	66	CC091	66	CC0B5	66	CC091	66	CC0B5
54	CC24E	68	002BC	66	CC0B5	86	CC0B5	86	CC0B5	86	CC0B5	86	CC0B5	86	CC0B5	86	CC0B5	86	CC0B5
80	CC2C5	88	00288	86	CC070	103	CC07F	103	CC070	103	CC07F	103	CC070	103	CC07F	103	CC070	103	CC07F
92	CC28E	110	00282	103	CC075	150	CC034	150	CC075	150	CC034	150	CC075	150	CC034	150	CC075	150	CC034
120	CC118	186	0024E	140	CC012	220	CC01C	220	CC012	220	CC01C	220	CC012	220	CC01C	220	CC012	220	CC01C
205	CC1B7	302	001FC	215	CC01F	402	CC014	402	CC01F	402	CC014	402	CC01F	402	CC014	402	CC01F	402	CC014
304	CC20E	407	00134	350	CC045	890	CC027	890	CC045	890	CC027	890	CC045	890	CC027	890	CC045	890	CC027
409	CC14C	940	002F7	852	CC070	998	CC025	998	CC070	998	CC025	998	CC070	998	CC025	998	CC070	998	CC025
951	CC307	1020	0021D	996	CC010	5050	CC014	5050	CC010	5050	CC014	5050	CC010	5050	CC014	5050	CC010	5050	CC014
5021	CC01C	6365	0015A	5023	CC02C														

LOCAL VARIABLES (1103 WORDS):

CC0C0 IBUF      CC038 NBM      CC0401 IGRCC      CC0402 IAREC  
 CC0C3 IPREC      CC0C4 IIN      CC0407 INIT      CC0408 PLNUM

00409 IND	0040A ITAPE	004CB NX	0040C NY	0040D NZ	0040E NM
0040F IDEC	00410 KPT	00411 KHT	00412 XFAC	00413 YFAC	00414 ZFAC
00415 WFAC	00416 ANGB	00417 XINC	00418 YINC	00419 ZZ	0041A ZMT
00418 KGT	0041C TBP	0041D BBT	0041E DLEFT	0041F RIGT	00420 TOPY
00421 BBTY	00422 DLEFX	00423 RIGTX	00424 LN	00425 IDEG	00426 AMIN
00427 RADN	00428 RTBP	00429 RBBT	0042A RLEFT	0042B RRIGHT	0042C BSVEC
0042C TSVEC	0042E IFLAG	0042F ISTDA	00430 RLEFT	00431 ISTYR	00432 ISTHM
00433 IENDA	00434 IENMB	00435 IENYR	0043C IENHM	00437 ISKP	00438 XORG
00439 YORG	0043A RFACT	0043B YT	0043C XT	0043D DATAX	0043E DATAY
0043F DATAZ	00440 DATAH	00441 RLAT	00442 RLONG	00443 JDA	00444 JMG
00445 JYR	00446 JHM	00447 IE9D	00448 XX	00449 YY	0044A KGDA
0044B KGM0	0044C KGYR	0044D KGHM	0044E INDK		

BLANK COMMON (0 WORDS)

EXTERNAL SUBPROGRAMS REQUIRED:

DMTOR	EXIT	FIND	GETG	GETL	GRIDG	ISM	NUMBER
PLOT	PLETS	SETSKP	SXPREC	SP0T	STAT	SYMBOL	TODAY
WHERE	F:101	F:102	F:103	F:104	F:105	F:106	F:108
MID0	M10C	9BCDREAD	9BCDWRIT	9ENDI0L	9INITIAL	9I0DATA	9I0LUSA
SITOR	9PRINT						

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	DEC	HEX
CONSTANTS:	WORDS	WORDS
LOCAL VARIABLES:	821	00335
TEMPS:	21	00015
	1103	0044F
	0	00000
	----	----
TOTAL PROGRAM:	1945	00799

```

1.  C      PROGRAM GRAV1
2.  C      OUTPUT (PROGRAM GRAV1, VERSION 7 APRIL 1975)
3.  C      MODIFIED BY C BROWN FROM BOUGUÉ VERSION OF 1 SEPT 1971
4.  C
5.  C
6.  C      PROGRAM GRAV1, REPRODUCES PRINCIPAL FACTS AT FORMAT OF 2 JULY 1969
7.  C      AND COMPLETES THE 132 CHARACTER GSUM FORMAT. OBSERVED
8.  C      GRAVITY IS CALCULATED IF ISW(3) = 1
9.  C
10. C      OUTPUT IS AT IGSN71 DATUM AND ANOMALIES CALCULATED WITH
11. C      INTERNATIONAL GRAVITY FORMULA OF 1967.
12. C
13. C
14. C      ASSUMED CRISTAL DENSITY IS ENTERED AT RUN TIME
15. C      RECALCULATES BOUGUER ANOMALY AT DENSITY ENTERED AT RUN TIME
16. C
17. C
18. C      DIMENSION IA(30)
19. C
20. C      SS*(1) = 1 FOR INPUT ELEV IN FEET
21. C      = 0 FOR INPUT ELEV IN METERS
22. C      SS*(2) = 1 FOR INPUT DEPTH IN FATHOMS
23. C      = 0 FOR INPUT DEPTH IN METERS
24. C      SS*(7) = 1 TO CALCULATE OBSERVED GRAVITY FROM FA ANOMALY
25. C      SS*(4) = 0 INPUT DATA IS ALREADY AT IGSN71 AND INT. GRAV FORM 1967
26. C      = 1 INPUT DATA AT PUTSOM SYSTEM AND INT GRAV FORM 1930
27. C      SS*(6) = 1 FOR INCORPORATING TERRAIN CORRECTION
28. C      = 0 FOR NOT USING TERRAIN CORRECTION
29. C      SS*(9) = 1 TO PRINT VALUES OF THEO AND FELEV
30. C      SS*(13) = 1 TO ASSIGN A STATION NUMBER, NUMERICALLY STARTING WITH
31. C      FIRST INPUT RECORD AS 1.
32. C      SS*(25) = 1 TO OUTPUT ON LINE PRINTER ONLY (GINST)
33. C      SS*(31) = 1 TO PUNCH GSUM OUTPUT ON 2 CARDS EACH (GINST)
34. C
35. C      IF DEPTH = 0, THEN PROGRAM ASSUMES DATA IS FOR A
36. C      LAND STATION AND USES ELEV IN BOUGUER COMPUTATION
37. C      OUTPUT ELEV AND DEPTH VALUES ARE IN METERS
38. C
39. C      USES STAT, GINST, GINTF, GI67F, ISW, NAVIN, RTDN2
40. C
41. C
42. C      *****
43. C      ITAPE = URN FOR INPUT
44. C      UTAPE = URN FOR OUTPUT
45. C      ITAPE = 1
46. C      UTAPE = 2
47. C      *****
48. C
49. C      IIN = 105
50. C      IIPUT = 105
51. C      CALL STAT
52. C      INIT = ISW(-2)
53. C      ACT = 0
54. C      ISN = 0
55. C      IRECP = 2
56. C      IDIF = 0
57. C      FA = 0.0
58. C      RG = 0.0
59. C      RFA = 0.0

```

```

60.      IREGC=0
61.      IFFC=0
62.      IFBC=0
63.      DB 1 J=1,35
64.      IA(J) = 1H
65.      1 CONTINUE
66.      C CRDEN = ASSUMED CRUSTAL DENSITY
67.      READ (IIN, 2) CRDEN
68.      2 FORMAT (F10.0)
69.      OUTPUT CRDEN
70.      C SET SS*(30) = 1 SO GINBT WILL NOT TRY TO READ EITP CARDS
71.      II = ICHG(30,1)
72.      KK=0
73.      CALL GINBT(ITAPE, JTAPE, KK, KGDA, KGM0,
74.      1 KGYR, KGHM, IDIF, ISORC, RLAT, RLONG, ELEV, K977, OBSG,
75.      2 IDEP, FA, BG, TC, IELC, IGC, RFA, IREGC, IFFC, IA, IFBC)
76.      50 CONTINUE
77.      52 READ (ITAPE, 14) ISORC, KGDA, KGM0, KGYR, KGHM, LAT,
78.      1 RLATM, KNS, LONG, RL0M, KEK, ELEV, K977, OBSG,
79.      2 IDEP, TC, IELC, IGC, AFA, ABG, ABGCM
80.      14 FORMAT (I4, 3I2, I4, I2, F5.2, A1, I3, F5.2, A1, F7.2,
81.      1 I3, F6.2, I5, F4.1, 2I2, 3F6.1)
82.      CALL STAT(1)
83.      CALL EVIL(IIOUT, I, IBAD, KGDA, KGM0, KGYR, KGHM)
84.      IF (IBAD) 50, 53, 55
85.      55 END FILE JTAPE
86.      WRITE(IIOUT, 54) NCT
87.      54 FORMAT ('END OF PROCESSING, NCT = ', I8)
88.      CALL EXIT
89.      53 CONTINUE
90.      IF (ISW(13).EQ.1) ISN = ISN+1, KGHM = ISN,
91.      1 KGDA = 0, KGM0 = 0, KGYR = 0
92.      70 DEPTH=IDEP
93.      CALL NAVIN(LAT, RLATM, KNS, LONG, RL0M, KEK, RLAT, RLONG)
94.      THEB=GINTE(RLAT)
95.      TH67 = GI67F(RLAT)
96.      DIF = (K977 - 977)*1000
97.      C
98.      IF (ISW(10)) 40, 42, 40
99.      C CONVERTING FEET TO METERS
100.      40 ELEV = ELEV * 0.304801
101.      42 FELEV = ((0.304855 + 0.00022*CB(2.*RLAT))*ELEV)
102.      1 * (((FELEV*0.001)**2.0)*0.072)
103.      FA=AFA
104.      IF (ISW(3).EQ.0) ZG0BS = OBSG + DIF, GO TO 505
105.      IF (ISW(4).EQ.0) ZG0BS = (AFA - FELEV) + TH67, GO TO 520
106.      C
107.      C CALCULATE OBSERVED GRAV IN 1930 IGF
108.      C
109.      PG0BS = (AFA - FELEV) + THEB
110.      GO TO 510
111.      505 IF (ISW(4).EQ.0) GO TO 550
112.      PG0BS = ZG0BS
113.      C
114.      C CHANGE TO IGSN 71
115.      510 ZG0BS = PG0BS - 14.0
116.      520 TG0BS = ZG0BS * 0.001
117.      IG0BS = TG0BS
118.      K977 = 977 + IG0BS

```

```

120.      BBSG = ZGBBS - A
121.      C
122.      C      CALCULATE FREE AIR ANOMALY
123.      C
124.      550  FA = (ZGBBS - TH67) + FELEV
125.      IF(ISW(8))43,45,43
126.      43  WRITE(11OUT,44)THE9,FELEV,TH67
127.      44  FORMAT(3F10.2)
128.      45  IF(ISW(2))46,200,46
129.      C  CONVERTING FATHOMS TO METERS
130.      46  DEPTH = DEPTH * 1.8288
131.      IDEP=DEPTH
132.      C  CHECKING IF WATER STATION
133.      200  CONTINUE
134.      IF(IDEP)165,160,165
135.      C  N9, ASSUME IT IS A LAND STATION
136.      160  DENS = CRDEN
137.      THICK = -ELEV
138.      GO TO 170
139.      C  YES, IT IS A WATER STATION
140.      165  DENS = CRDEN = 1.03
141.      THICK = DEPTH
142.      170  BG = FA + (0.04185*DENS*THICK)
143.      C  INCORPORATE TERRAIN CORRECTION
144.      IF(ISW(6))76,74,76
145.      74  TC=99.9
146.      BGC0M=999.0
147.      GO TO 80
148.      76  IF(TC=99.7)79,78,78
149.      78  BGC0M=999.0
150.      GO TO 80
151.      79  BGC0M = BG + TC
152.      C  ROUNDING BGC0M
153.      IF(BGC0M)279,280,280
154.      279  BGC0M=BGC0M-0.05
155.      GO TO 80
156.      280  BGC0M=BGC0M+0.05
157.      C  ROUNDING OF FREE-AIR
158.      80  IF(FA)201,220,220
159.      201  FA=FA-0.05
160.      GO TO 250
161.      220  FA=FA+0.05
162.      C  ROUNDING OF BOUGUER
163.      250  IF(BG)255,260,260
164.      255  BG=BG-0.05
165.      GO TO 380
166.      260  BG=BG+0.05
167.      380  CONTINUE
168.      KK=-2
169.      CALL      GINBT(ITAPE,UTAPE,KK,KGDA,KGM8,
170.      1  KGYR,KGMM,IDIF,ISORC,RLAT,RLONG,ELEV,K977,BBSG,
171.      2  IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
172.      NCT=NCT+1
173.      GO TO 50
174.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	0004C	1	ABGM	R	SCALR	0004C	1	ABGM	R	SCALR	0004C	1
AF	R	SCALR	0005A	1	BGCM	R	SCALR	0005A	1	BGCM	R	SCALR	0005A	1
CS	R	SPRGG	0006A	1	CRDN	R	SCALR	0006A	1	DENS	R	SCALR	00059	1
DEPTH	R	SCALR	0004F	1	DIF	R	SCALR	00033	1	ELEV	R	SCALR	0003D	1
EVIL	R	SPRGG	0004F	1	EXTN	R	SPRGG	0004F	1	FA	R	SCALR	0002C	1
FELEV	R	SCALR	00033	1	GIN I	R	SPRGG	0004F	1	GINIF	R	SPRGG	0002C	1
GI67F	R	SPRGG	0004F	1	ICH3	R	SCALR	0004C	1	IA	R	ARRAY	00000	35
IBAC	R	SCALR	0004F	1	IELC	R	SCALR	00042	1	IDEP	R	SCALR	00040	1
IDIF	R	SCALR	00029	1	IGC	R	SCALR	00043	1	IFBC	R	SCALR	00031	1
IFFC	R	SCALR	0003A	1	IIN	R	SCALR	00043	1	IG8BS	R	SCALR	00057	1
II	R	SCALR	0003A	1	IIR	R	SCALR	00025	1	IIBUT	R	SCALR	00026	1
INIT	R	SCALR	00027	1	IIR2	R	SCALR	0002A	1	IIRGC	R	SCALR	0002F	1
ISN	R	SCALR	00029	1	ISB5C	R	SCALR	0003A	1	ISW	R	SPRGG	0002F	1
ITAPE	R	SCALR	00029	1	J	R	SCALR	00032	1	JTAPE	R	SCALR	00024	1
KEM	R	SCALR	00049	1	K3C4	R	SCALR	00036	1	KGMM	R	SCALR	00039	1
KN5	R	SCALR	00037	1	KGR	R	SCALR	00038	1	KK	R	SCALR	00035	1
LONG	R	SCALR	00047	1	K977	R	SCALR	0003E	1	LAT	R	SCALR	00044	1
OB50	R	SCALR	0003F	1	NAVIN	R	SPRGG	0004F	1	NCT	R	SCALR	00028	1
RLAT	R	SCALR	0003F	1	PG85S	R	SCALR	00055	1	RFA	R	SCALR	0002E	1
RLNG3	R	SCALR	0003C	1	RLATM	R	SCALR	00045	1	RLBM	R	SCALR	00048	1
TL8BS	R	SCALR	00056	1	STAT	R	SPRGG	0004F	1	TC	R	SCALR	00041	1
TH67	R	SCALR	00051	1	TH67	R	SCALR	00054	1	THICK	R	SCALR	0005A	1
					Z68PS	R	SCALR	00054	1					

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
1	R	SCALR	00036	1	00085	R	SCALR	00085	1	00026	R	INIT	00027	1
44	R	SCALR	0013A	1	46	R	SCALR	00142	1	0002C	R	FA	0002D	1
57	R	SCALR	000AC	1	70	R	SCALR	00066	1	00031	R	IFRC	00032	1
79	R	SCALR	0016A	1	160	R	SCALR	0014A	1	00037	R	KGR	00039	1
201	R	SCALR	00178	1	250	R	SCALR	0017F	1	0003E	R	K977	0003F	1
280	R	SCALR	00173	1	505	R	SCALR	00110	1	00043	R	LAT	00045	1

## LOCAL VARIABLES (92 WORDS)

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
00000	R	SCALR	00023	1	00024	R	JTAPE	00024	1	00025	R	IFN	00025	1
00028	R	SCALR	00029	1	0002A	R	IFEC2	0002A	1	0002B	R	IOIF	0002B	1
0002E	R	SCALR	0002F	1	0002C	R	IFRC	0002C	1	00031	R	IFRC	00031	1
0003A	R	SCALR	0003B	1	00036	R	KGR	00036	1	00037	R	KGR	00037	1
00040	R	SCALR	00041	1	00042	R	IELC	00042	1	00043	R	IELC	00043	1
00046	R	SCALR	00047	1	00047	R	IELC	00047	1	00049	R	KEA	00049	1
0004C	R	SCALR	0004D	1	0004E	R	IELC	0004E	1	0004F	R	KEA	0004F	1
00052	R	SCALR	00053	1	00053	R	FELEV	00053	1	00055	R	PG8BS	00055	1
00058	R	SCALR	00059	1	00059	R	DEVS	00059	1	0005B	R	PGCM	0005B	1

## BLANK COMMON (0 WORDS)

## INTRINSIC SUBPROGRAMS USED:

C85

## EXTERNAL SUBPROGRAMS REQUIRED:

EVIL	EXIT	GINBT	GINTE	GI67F	ICMG	ISW	NAVIN
STAT	F:101	F:102	F:103	F:104	F:105	F:106	F:108
MID9	M:1C	9BCOREAD	9BCDARIT	9C6S	9ENDFILE	9ENDI6L	9INITIAL
9I0DATA	9IT6R	9PRINT	9PWRRR	95T6I			

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	423	001A7
CONSTANTS:	36	0001A
LOCAL VARIABLES:	22	0005C
TEMPS:	1	00001
TOTAL PROGRAM:	512	0021E

```

1.  C      PROGRAM GST0G
2.  C      PROGRAM GST0G,  CONVERTS SEAG1 FORMAT TO GSUM FORMAT
3.  C
4.  C      VERSION 25 JUNE 75,  TO ADD OUTPUT STATEMENT
5.  C      VERSION 13 JAN 75 TO READ IREC#2
6.  C      VERSION 0F 4 OCT 1971,  TO USE GINOT FOR OUTPUT
7.  C      VERSION 0F 20 JANUARY 1971
8.  C
9.  C      SIGMA 7 VERSION HANDLES SINGLE REEL INPUT ONLY
10. C
11. C      SSW(12) UP TO LIST DATA IDENTIFICATION
12. C
13. C      USES BIC0R, STAT, ISW, EVIL
14. C
15. C      CALLS STAT, MCV0L, GETDCB FROM ACCOUNT 3 LIBRARY
16. C
17. C
18. C      DIMENSION MAG(2)
19. C      DIMENSION MAG1(5),MAG2(5),IBLK(21)
20. C      DIMENSION IA(35),IZ(9),IW(35)
21. C      DIMENSION N0W(4)
22. C      IIN = 105
23. C      IIBUT = 108
24. C
25. C      PRINT DATE AND TIME 0F JOB 0N HEADING
26. C      CALL T0DAY(N0W)
27. C      WRITE(IIBUT,13) N0W
28. C      13  FORMAT(1X,4A4)
29. C
30. C      OUTPUT : GST0G VERSION 0F 25 JUNE 75:
31. C
32. C      215  IREC1=1
33. C      FOR MULTIFILE MAGTAPE INPUT
34. C      NEF=1
35. C      FOR WRITING EOF 0N JTAPE
36. C      IFLAG=1
37. C      CALL STAT
38. C      INN = ISW(-2)
39. C      NZERO=0
40. C      KGDA0 = NZERO
41. C      KGM00=NZERO
42. C      KGYR0=NZERO
43. C      KGHM0=NZERO
44. C
45. C      *****
46. C      ITAPE = INPUT TAPE
47. C      JTAPE = OUTPUT TAPE
48. C      ITAPE = 1
49. C      JTAPE = 2
50. C      *****
51. C
52. C      IDIF=0
53. C      ELEV=0.0
54. C      TC=99.9
55. C      RFA=0.0
56. C      IREGC=0
57. C      IFFC=7
58. C      IFBC=0

```

```

60.      NREJ = 0
61.      ICHECK = 0
62.      NOUT=0
63.      DEGRA=1.745329E-2
64.      C
65.      C      ISORC = SOURCE NUMBER
66.      C      IDCBD = 0 FOR ID BY DATE, 1 FOR ID BY STA NO.
67.      C      IELC = ELEVATION CODE
68.      C      IGC = G-METER CODE
69.      C      BIAS = MGAL BIAS
70.      C      NFILE = NUMBER OF FILES TO BE INPUT
71.      C
72.      READ (IIN,2) ISORC, IDCBD, IELC, IGC, BIAS
73.      2      FORMAT (4I5, F10.0)
74.      OUTPUT ISORC, IDCBD, IELC, IGC, BIAS
75.      READ (IIN, 5) NFILE
76.      5      FORMAT ( I5)
77.      WRITE(IOUT, 7786) NFILE
78.      7786      FORMAT(IGST06: NFILE=I, I5)
79.      IF(IDCBD)230,226,230
80.      226      KGDA=0
81.      KGM0=0
82.      KGYR=0
83.      C
84.      C
85.      230      CONTINUE
86.      KK=0
87.      CALL      GIN0T(ITAPE,UTAPE,KK,KGDA,KGM0,
88.      1      KGYR,KGHM,IDIF,ISORC,RLAT,RLONG,ELEV,K977,0BSG,
89.      2      IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
90.      C
91.      C
92.      C
93.      C      READ INPUT DATA
94.      50      CONTINUE
95.      52      READ(ITAPE,360)IREC1,KGDA,KGM0,KGYR,KGHM,IDIF,
96.      1      RLAT,RLONG,KVN,KVE,K977,I0GR,KFA,KBG,KCVN,
97.      2      KCVE,KCDM,MTDC,MT,MAG(1),MAG(2),KETV0
98.      360      FORMAT(I1,3I2, I4,I3,2F9.6,2I5,I3,I4,5I5,I3,
99.      1      I2,I1,I4,I5)
100.      CALL STAT(I)
101.      CALL EVIL(IOUT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0)
102.      C      IBAD = -, PARITY OR FORMAT ERROR
103.      C      IBAD = 0, READ WAS OK
104.      C      IBAD = +, END OF FILE OR END OF TAPE ENCOUNTERED
105.      IF (IBAD) 50, 53, 575
106.      53      IF (IREC1.EG.2) GO TO 70
107.      IF (IREC1.NE. 1) GO TO 60
108.      IF (ICHECK.EG.1) GO TO 70
109.      BLTPUT'***** INPUT IS IN 1930 IGF *****'
110.      ICHECK = 1
111.      GO TO 70
112.      60      IF (IREC1-9)50,62,50
113.      62      READ(ITAPE,64)IREC9,I2,I21,ITEST
114.      64      FORMAT(I1,2I3,I4)
115.      IF (ITEST-6563)580,565,580
116.      565      WRITE(IOUT,570)
117.      570      FORMAT(IE0R)
118.      CS      PAUSE 400
119.      CS      GO TO 50

```

```

120. C      EVIL HAS FOUND AN END OF FILE MARK ON INPUT
121. 575    IF (NEF . NE. 1) 576, 577, 577
122. 576    NEF = NEF + 1
123.        CALL MCVOL (ITAPE)
124.        GO TO 50
125. C      END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
126. 577    IF (IFLAG.NE.1) GO TO 578
127.        END FILE JTAPE
128. 578    WRITE (IIOUT, 579) NREJ, NOUT
129. 579    FORMAT ('END OF PROCESSING', //
130. 1 'NUMBER OF RECORDS REJECTED BECAUSE OF INVALID GRAVITY = ', I6
131. 2 //, 'NUMBER OF RECORDS OUTPUT = ', I6)
132.        CALL EXIT
133. 580    IF (ITEST-6665) 68, 65, 68
134. 65    WRITE (IIOUT, 601)
135. 601    FORMAT ('EOD')
136.        GO TO 575
137. 68    WRITE (IIOUT, 69)
138. 69    FORMAT ('IREC1=9')
139.        GO TO 50
140. 70    CONTINUE
141.        KGDA0=KGDA
142.        KGM00=KGM0
143.        KGYR0=KGYR
144.        KGHM0=KGHM
145. C
146. C      CHECKING GRAVITY FOR INVALID DATA
147. C
148.        IF (K977.LE.0) NREJ = NREJ + 1; GO TO 50
149.        IF (KFA.GE.9900) NREJ = NREJ + 1; GO TO 50
150.        OMSG=FLOAT(I0GR)*0.1
151.        FA=FLOAT(KFA)*0.1
152.        BG=FLOAT(KBG)*0.1
153.        IDEP=KCDM
154.        CALL BICOR(K977, OMSG, BIAS)
155.        IF (FA-990.0) 85, 87, 87
156. 85    FA=FA+BIAS
157. 87    IF (BG-990.0) 88, 90, 90
158. 88    BG=BG+BIAS
159. 90    CONTINUE
160. C      GROUPING VARIABLES FOR OUTPUT UNDER ARRAY IA
161.        ENCODE(35, 410, IZ) KVN, KVE, KCVN, KCVE, KETV0, MTDC, MT
162. 410    FORMAT(4I5, I5, I3, I2, 5X)
163.        CALL UNPKBY (IZ, IW, 35)
164.        DO 420 J=1, 35
165.        IA(J)=ISL(IW(J), 24)
166. 420    CONTINUE
167. C
168.        KK=-2
169.        CALL GINOT (ITAPE, JTAPE, KK, KGDA, KGM0,
170. 1 KGYR, KGHM, IDIF, ISORC, RLAT, RLONG, ELEV, K977, OMSG,
171. 2 IDEP, FA, BG, TC, IELC, IGC, RFA, IREGC, IFFC, IA, IFBC)
172.        NOUT=NOUT+1
173.        GO TO 50
174. C      WRITE END OF FILE RECORD
175. 999    END FILE JTAPE
176.        WRITE (IIOUT, 330) KGDA, KGM0, KGYR, KGHM
177. 330    FORMAT ('DATA BEFORE EOR = ', 3I3, I5)
178.        CALL EXIT
179. END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
DEGRA	R	SCALR	0009E V	1	BIAS	R	SCALR	00091 V	1	BICOR	R	SCALR	00091 V	1	ICHECK	R	SCALR	00021 V	35
EXIT	R	SCALR	0008C V	1	ELEV	R	SCALR	00082 V	1	EVIL	R	SCALR	00082 V	1	IDIF	R	SCALR	00081 V	1
GINBT	R	SPRGG	0008A V	1	FA	R	SCALR	0009D V	1	INIRIN	R	SPRGG	0009D V	1	IFFC	R	SCALR	00086 V	1
IBAD	R	SPRGG	0008B V	1	IBLK	R	SCALR	000AA V	21	IA	R	ARRAY	000AA V	1	IIN	R	SCALR	00074 V	1
ICD8D	R	SCALR	0008E V	1	IDEP	R	SCALR	000CC V	1	ICHECK	R	SCALR	000CC V	1	IIRGR	R	SCALR	000A1 V	1
IELC	R	SCALR	0008F V	1	IFBC	R	SCALR	0009C V	1	IFFC	R	SCALR	0009C V	1	IISW	R	SCALR	00085 V	1
IFLAG	R	SCALR	0008F V	1	IGC	R	SCALR	00090 V	1	IIN	R	SCALR	00090 V	1	IISW	R	SCALR	00085 V	1
IBOUT	R	SCALR	00078 V	1	INN	R	SCALR	00097 V	1	IIRGR	R	SCALR	00097 V	1	IISW	R	SCALR	00085 V	1
IREC1	R	SCALR	00076 V	1	ISORC	R	SCALR	000AC V	1	IIRGR	R	SCALR	000AC V	1	IISW	R	SCALR	00085 V	1
ISL	R	SPRGG	00076 V	1	ITEST	R	SCALR	000AD V	1	IISW	R	SCALR	000AD V	1	IISW	R	SCALR	00085 V	1
ITAPE	R	SCALR	0007F V	1	IZ1	R	SCALR	000AD V	1	IISW	R	SCALR	000AD V	1	IISW	R	SCALR	00085 V	1
IZ	R	ARRAY	00044 V	9	KBG	R	SCALR	000A3 V	1	IISW	R	SCALR	000A3 V	1	IISW	R	SCALR	00085 V	1
UTABE	R	SCALR	00080 V	1	KCVN	R	SCALR	000A4 V	1	IISW	R	SCALR	000A4 V	1	IISW	R	SCALR	00085 V	1
KCVC	R	SCALR	000A5 V	1	KGDA	R	SCALR	000A4 V	1	IISW	R	SCALR	000A4 V	1	IISW	R	SCALR	00085 V	1
KFA	R	SCALR	000A2 V	1	KGMH	R	SCALR	000A2 V	1	IISW	R	SCALR	000A2 V	1	IISW	R	SCALR	00085 V	1
KGMH	R	SCALR	00097 V	1	KGYR	R	SCALR	0007E V	1	IISW	R	SCALR	0007E V	1	IISW	R	SCALR	00085 V	1
KGMH	R	SCALR	0007C V	1	KVE	R	SCALR	00095 V	1	IISW	R	SCALR	00095 V	1	IISW	R	SCALR	00085 V	1
KK	R	SCALR	00096 V	1	KV	R	SCALR	000A0 V	1	IISW	R	SCALR	000A0 V	1	IISW	R	SCALR	00085 V	1
K977	R	SCALR	0009A V	1	MAG	R	SCALR	000C0 V	1	IISW	R	SCALR	000C0 V	1	IISW	R	SCALR	00085 V	1
MAG2	R	ARRAY	00007 V	5	MCVBL	R	SCALR	000C0 V	1	IISW	R	SCALR	000C0 V	1	IISW	R	SCALR	00085 V	1
MTDC	R	SCALR	00007 V	1	NEF	R	SCALR	00077 V	1	IISW	R	SCALR	00077 V	1	IISW	R	SCALR	00085 V	1
NBLT	R	SCALR	00007 V	1	NBL	R	SCALR	00070 V	4	IISW	R	SCALR	00070 V	1	IISW	R	SCALR	00085 V	1
NREJ	R	SCALR	0008B V	1	NZRB	R	SCALR	0007A V	1	IISW	R	SCALR	0007A V	1	IISW	R	SCALR	00085 V	1
RFA	R	SCALR	00089 V	1	RLAT	R	SCALR	0007A V	1	IISW	R	SCALR	0007A V	1	IISW	R	SCALR	00085 V	1
STAT	R	SPRGG	00084 V	1	TC	R	SCALR	00098 V	1	IISW	R	SCALR	00098 V	1	IISW	R	SCALR	00085 V	1
UNPRBY	R	SPRGG	00084 V	1				00083 V	1	IISW	R	SCALR	00083 V	1	IISW	R	SCALR	00085 V	1

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
2	00059	5	0007C	13	0000E	50	000AE
60	000FF	62	00102	64	0010D	65	00152
70	00163	85	0018A	87	0018D	88	00190
226	0008B	230	00051	330	0010B	360	000C8
565	00114	570	00118	575	0011B	576	0011E
579	0012E	580	0014F	601	00156	999	001D1

## LOCAL VARIABLES (176 WORDS):

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
0000C MAG	00002	00070 MAG1	00002	00077 MAG2	00074	00075 MAG2	00074
0004C IM	00070	00070 NAW	00070	00074 IN	00074	00075 IN	00074
00078 IFLAG	00079	00079 IN	00079	00074 IN	00074	00075 IN	00074
0007E KGPB	0007F	0007F ITAF	0007F	00074 IN	00074	00075 IN	00074
00084 RFA	00085	00085 IREGC	00085	00074 IN	00074	00075 IN	00074
0008A ICHECK	0008F	0008F NBT	0008F	00074 IN	00074	00075 IN	00074
00090 IGC	00091	00091 BIAS	00091	00074 IN	00074	00075 IN	00074
00096 KK	00097	00097 KGMH	00097	00074 IN	00074	00075 IN	00074
0009C IDEP	0009D	0009D FA	0009D	00074 IN	00074	00075 IN	00074
000A2 KFA	000A3	000A3 KGB	000A3	00074 IN	00074	00075 IN	00074
000A8 MT	000A9	000A9 KETV8	000A9	00074 IN	00074	00075 IN	00074
000AC ITES	000AF	000AF J	000AF	00074 IN	00074	00075 IN	00074

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
53	000E5	53	000E5	53	000E5	53	000E5
69	0015E	69	0015E	69	0015E	69	0015E
215	0001E	215	0001E	215	0001E	215	0001E
420	00180	420	00180	420	00180	420	00180
578	00128	578	00128	578	00128	578	00128

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
00044 IZ	00044	00077 NEF	00077	0007D KGMH8	0007D	00083 TC	00083
00077 NEF	00077	00083 TC	00083	00089 NREJ	00089	0008F IELC	0008F
0007D KGMH8	0007D	00089 NREJ	00089	00095 KGYR	00095	00098 BBSG	00098
00083 TC	00083	00095 KGYR	00095	0009A K977	0009A	000A1 IGR	000A1
00089 NREJ	00089	0009A K977	0009A	000A0 KVN	000A0	000A7 MTDC	000A7
00095 KGYR	00095	000A0 KVN	000A0	000A6 KCDM	000A6	000AD IZ1	000AD
00098 BBSG	00098	000A6 KCDM	000A6	000AC IREC9	000AC		

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

FL0AT ISL

EXTERNAL SUBPROGRAMS REQUIRED:

BIC0R	EVIL	EXIT	GIN0T	ISW	MCV0L	STAT	T0DAY
UNPKEY	F:101	F:102	F:103	F:104	F:105	F:106	F:108
M:00	M:0C	9BCDREAD	9BCDWRT	9ENC0DE	9ENDFILE	9ENDI0L	9INITIAL
9I0DATA	9I0LLSA	9IT0R	9PRINT	9STOP			

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	487	C01E7
CONSTANTS:	6	C0C06
LOCAL VARIABLES:	176	C0C00
TEMPS:	0	C0C0C
TOTAL PROGRAM:	669	C029D

```

1.  C   PROGRAM GSTGG67
2.  C   PROGRAM GSTGG67 MOD OF GSTGG TO GO 1967 FORULA AND 71 DATUM
3.  C   CHANGE WHERE NECESSARY
4.  C   MOD FROM GSTGG OF 25 JUNE 75
5.  C   PROGRAM GSTGG, CONVERTS SEAG1 FORMAT TO GSUM FORMAT
6.  C
7.  C   VERSION OF 10 MAR 76 TO FIX IRECC FOR I/P AT 67
8.  C   VERSION OF 5 AUG 75 TO DO 67 FORMULA CONVERSION
9.  C   AND TO WRITE GSUM O/P BLOCKED BY 50
10. C   OUTPUT 'GSTGG67 VERSION 10 MAR 76'
11. C   VERSION 25 JUNE 75, TO ADD OUTPUT STATEMENT
12. C   VERSION 13 JAN 75 TO READ IREC#2
13. C   VERSION OF 4 OCT 1971, TO USE GINOT FOR OUTPUT
14. C   VERSION OF 20 JANUARY 1971
15. C
16. C   SIGMA 7 VERSION HANDLES SINGLE REEL INPUT ONLY
17. C
18. C   SSW(12) UP TO LIST DATA IDENTIFICATION
19. C
20. C
21. C
22. C
23. C   DIMENSION IBUFOT(32,50)
24. C   DIMENSION MAG(2)
25. C   DIMENSION MAG1(5),MAG2(5),IBLK(21)
26. C   DIMENSION IA(35),IZ(9),IW(35)
27. C   DIMENSION NOW(4)
28. C   IIN = 105
29. C   IIOUT = 108
30. C   KI=1
31. C   KB=2
32. C
33. C   PRINT DATE AND TIME OF JOB ON HEADING
34. C   CALL TODAY(NOW)
35. C   WRITE(IIOUT,13) NOW
36. C   13  FORMAT(1X,4A4)
37. C
38. C   RADEG=57.2958
39. C
40. C   215  IREC1=1
41. C   FOR MULTIFILE MAGTAPE INPUT
42. C   NEF=1
43. C   FOR WRITING EOF ON JTAPE
44. C   IFLAG=1
45. C   NROUT=0
46. C   INN = ISW(-2)
47. C   IKEY=0
48. C   NIN=0
49. C   NZERO=0
50. C   KGDAB = NZERO
51. C   KGMBO=NZERO
52. C   KGYRO=NZERO
53. C   KGHMO=NZERO
54. C
55. C   *****
56. C   JTAPE = INPUT TAPE
57. C   JTAPE = OUTPUT TAPE
58. C   JTAPE = 1
59. C   JTAPE = 2

```

```

60. C *****
61. C
62. IDIF=0
63. ELEV=0.0
64. IREC=2
65. TC=99.9
66. RFA=0.0
67. IREGC=0
68. IFFC=7
69. IFBC=0
70. NRECT=0
71. KK=0
72. NREJ = 0
73. ICHECK = 0
74. NOUT=0
75. DEGRA=1.745329E=2
76. C
77. C ISRC = SOURCE NUMBER
78. C IDCOD = 0 FOR ID BY DATE, 1 FOR ID BY STA NO.
79. C IELC = ELEVATION CODE
80. C IGC = G-METER CODE
81. C BIAS = MGAL BIAS
82. C NFILE = NUMBER OF FILES TO BE INPUT
83. C
84. READ (IIN,2) ISRC, IDCOD, IELC, IGC, BIAS
85. 2 FORMAT (4I5, F10.0)
86. OUTPUT ISRC, IDCOD, IELC, IGC, BIAS
87. READ (IIN, 5) NFILE
88. 5 FORMAT ( I5)
89. WRITE(IIOU, 7786) NFILE
90. 7786 FORMAT(IGSTOGI NFILE=,I5)
91. IF(IDCOD)230,226,230
92. 226 KGDA=0
93. KGM=0
94. KGYR=0
95. C
96. C
97. 230 CONTINUE
98. C
99. C
100. C
101. C READ INPUT DATA
102. 50 CONTINUE
103. 52 READ(ITAPE,360,END=575,ERR=50)IREC1,KGDA,KGM,KGYR,KGHM,IDIF,
104. 1 RLAT,RLONG,KVN,KVE,K977,IGR,KFA,KBG,KCVN,
105. 2 KCVE,KCDM,MTDC,MT,MAG(1),MAG(2),KETV0
106. 360 FORMAT(I1,3I2, I4,I3,2F9.6,2I5,I3,I4,5I5,I3,
107. * I2,I1,I4,I5)
108. IF(IREC1.EQ.2) GO TO 70
109. IF(IREC1.NE.1) OUTPUT 'IREC NE 1 OR 2'OUTPUT NREJ,NROUT,NOUT,STOP
110. C CONVERSION TO 67
111. IGSS=(K977*10000)*IGR
112. IGSS=IGSS-140
113. K977=IGSS/10000
114. IGR=IGSS-K977*10000
115. DG=3.2*(13.6*(SIN(ABS(RLAT))*2))
116. KDQ=FIX(DG*10.0)
117. IF(KFA.LT.9990) KFA=KFA+KDQ
118. IF(KBG.LT.9990) KBG=KBG+KDQ
119. GO TO 70

```

```

120. C          END OF FILE
121. 575 IF (NEF = NFILE) 576, 577, 577
122. 576 NEF = NEF + 1
123.      CALL MCVOL (JTAPE)
124.      GO TO 50
125. C          END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
126. 577 IF (IFLAG.NE.1) GO TO 578
127. C          LAST BUFF OUT
128.      IF (NROUT.EQ.0) GO TO 1577
129.      CALL BUFFER OUT (JTAPE, 0, IBUFST(1,1), NROUT*32)
130. 1577 CONTINUE
131.      END FILE JTAPE
132. 578 WRITE (IIOUT, 579) NREJ, NOUT
133. 579 FORMAT ('END OF PROCESSING', /,
134. 1 'NUMBER OF RECORDS REJECTED BECAUSE OF INVALID GRAVITY =', I6,
135. 2 /, 'NUMBER OF RECORDS OUTPUT =', I6)
136.      CALL EXIT
137. 580 IF (ITEST=6665) 68, 65, 68
138. 65 WRITE (IIOUT, 601)
139. 601 FORMAT ('EOD')
140.      GO TO 575
141. 68 WRITE (IIOUT, 69)
142. 69 FORMAT ('IREC1=9')
143.      GO TO 50
144. 70 CONTINUE
145.      KGDA0=KGDA
146.      KGM00=KGM0
147.      KGYR0=KGYR
148.      KGHM0=KGHM
149. C
150. C CHECKING GRAVITY FOR INVALID DATA
151. C
152.      IF (K977.LE.0) NREJ = NREJ + 1, GO TO 50
153.      IF (KFA.GE.9900) NREJ = NREJ + 1, GO TO 50
154.      DLAT=RLAT*RADEG
155.      PLAT=DLAT+90./LTKEY=PLAT
156.      DLONG=RLONG*RADEG
157.      PLONG=DLONG+180./LGKEY=PLONG
158.      BBSG=FLOAT (IBGR)*0.1
159.      FA=FLOAT (KFA)*0.1
160.      BG=FLOAT (KBG)*0.1
161.      IDEP=KCDM
162.      CALL BICOR (K977, BBSG, BIAS)
163.      IF (FA=990.0) 85, 87, 87
164. 85 FA=FA+BIAS
165. 87 IF (BG=990.0) 88, 90, 90
166. 88 BG=BG+BIAS
167. 90 CONTINUE
168. C GROUPING VARIABLES FOR OUTPUT UNDER ARRAY IA
169.      ENCODE (35, 410, IZ) KVN, KVE, KCVN, KCVE, KETVB, MTDC, MT
170. 410 FORMAT (4I5, I5, I3, I2, 5X)
171.      CALL UNPKBY (IZ, IW, 35)
172.      DO 420 J=1, 35
173.      IA(J)=ISL (IW(J), 24)
174. 420 CONTINUE
175. C
176.      IF (NROUT.EQ.50) CALL BUFFER OUT (JTAPE, 0, IBUFST(1,1), 1600, JKEY)
177.      *      NROUT=0
178.      NROUT=NROUT+1
179.      ENCODE (128, 1020, IBUFST(1, NROUT), NE)

```

```

180.      *      IREC,ISORC,KGDA,KGMB,KGYR,KGHM,DLAT,DLONG,ELEV,
181.      *      K977,OBSSG,IDEF,FA,BG,TC,IELC,IGC,RFA,IREGC,
182.      *      IFFC,IA,IFBC,LYKEY,LGKEY,IAKEY
183.      1020 FORMAT(I1,I4,3I2,I4,2F9.4,F7.2,I3,F6.2,I5,2F6.1,F4.1,
184.      *      2I2,F6.1,I1,I2,35A1,1X,I1,2I3,I2)
185.      NOUT=NOUT+1
186.      GO TO 50
187.      C      WRITE END OF FILE RECORD
188.      999      END FILE JTAPE
189.      WRITE(I1OUT,330)KGDA,KGMB,KGYR,KGHM
190.      330      FORMAT('DATA BEFORE EOM ',3I3,I5)
191.      CALL EXIT
192.      END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
ABS	R	SPRGG	0000	1	BG	R	SCALR	00F9	1	BIAS	R	SCALR	00D9	1	BIAS	R	SCALR	00D9	1
BICOR	R	SPRGG	0001	1	BUFFERBU	R	SPRGG	00F0	1	DEGRA	R	SCALR	00D4	1	DEGRA	R	SCALR	00D4	1
DO	R	SCALR	0002	1	EXIT	R	SPRGG	00F1	1	DLNG	R	SCALR	00F8	1	DLNG	R	SCALR	00F8	1
ELEV	R	SPRGG	0003	1	IA	I	ARRAY	0061	35	FA	I	SCALR	00E8	1	FA	I	SCALR	00E8	1
FLOAT	R	SPRGG	0004	1	IBUFOT	I	ARRAY	0000	1600	IAKEY	I	SCALR	00E6	1	IAKEY	I	SCALR	00E6	1
IBLK	I	SCALR	0005	1	IDEP	I	SCALR	00FA	1	ICHECK	I	SCALR	00D2	1	ICHECK	I	SCALR	00D2	1
IOGRO	I	SCALR	0006	1	IFSC	I	SCALR	00CE	1	IDIF	I	SCALR	00C7	1	IDIF	I	SCALR	00C7	1
ILC	I	SCALR	0007	1	IFLAG	I	SCALR	00B8	1	IFFC	I	SCALR	00C0	1	IFFC	I	SCALR	00C0	1
IFIX	I	SPRGG	0008	1	IN	I	SCALR	00B4	1	IGC	I	SCALR	00D8	1	IGC	I	SCALR	00D8	1
IGRBS	I	SCALR	0009	1	INR	I	SCALR	00E4	1	ITOUT	I	SCALR	00B5	1	ITOUT	I	SCALR	00B5	1
INN	I	SCALR	0010	1	IREGC	I	SCALR	00E4	1	IRL	I	SCALR	00C9	1	IRL	I	SCALR	00C9	1
INECI	I	SCALR	0011	1	ISW	I	SCALR	00CC	1	ISL	I	SPRGG	INTRIN	1	ISL	I	SPRGG	INTRIN	1
ISORC	I	SCALR	0012	1	ISW	I	SPRGG	EXTRN	35	ITAPE	I	SCALR	00C5	1	ITAPE	I	SCALR	00C5	1
ITEST	I	SCALR	0013	1	J	I	ARRAY	008D	35	IZ	I	ARRAY	00B4	1	IZ	I	ARRAY	00B4	1
J	I	SCALR	0014	1	JKEY	I	SCALR	00FC	1	JTAP	I	SCALR	00C6	1	JTAP	I	SCALR	00C6	1
KBG	I	SCALR	0015	1	KCDH	I	SCALR	00E9	1	KVE	I	SCALR	00E9	1	KVE	I	SCALR	00E9	1
KCVN	I	SCALR	0016	1	KOG	I	SCALR	00E7	1	KETV8	I	SCALR	00EC	1	KETV8	I	SCALR	00EC	1
KFA	I	SCALR	0017	1	KGDA	I	SCALR	00DB	1	KGDAB	I	SCALR	00C1	1	KGDAB	I	SCALR	00C1	1
KGMH	I	SCALR	0018	1	KGMH8	I	SCALR	00C4	1	KGM8	I	SCALR	00C3	1	KGM8	I	SCALR	00C3	1
KGMH8	I	SCALR	0019	1	KGYR	I	SCALR	00DD	1	KGYR8	I	SCALR	00C3	1	KGYR8	I	SCALR	00C3	1
KI	I	SCALR	0020	1	KK	I	SCALR	00DD	1	K9	I	SCALR	00E7	1	K9	I	SCALR	00E7	1
KVE	I	SCALR	0021	1	KVN	I	SCALR	00E0	1	K977	I	SCALR	00E3	1	K977	I	SCALR	00E3	1
LGKEY	I	SCALR	0022	1	LTKEY	I	SCALR	00E1	1	MAG	I	ARRAY	0040	2	MAG	I	ARRAY	0040	2
MAG1	I	ARRAY	0023	1	MAG2	I	ARRAY	0047	5	MCVBL	I	SPRGG	EXTRN	1	MCVBL	I	SPRGG	EXTRN	1
MT	I	SCALR	0024	1	MTDC	I	SCALR	00E4	1	NE	I	SCALR	00FD	1	NE	I	SCALR	00FD	1
NEF	I	SCALR	0025	1	NFILE	I	SCALR	00DA	1	NIN	I	SCALR	00BF	1	NIN	I	SCALR	00BF	1
NRT	I	SCALR	0026	1	N8H	I	ARRAY	0080	4	NRECT	I	SCALR	00C7	1	NRECT	I	SCALR	00C7	1
NREJ	I	SCALR	0027	1	NROUT	I	SCALR	00B0	1	NZER8	I	SCALR	00C0	1	NZER8	I	SCALR	00C0	1
88SG	R	SCALR	0028	1	PLAT	R	SCALR	00B2	1	PLNG	R	SCALR	00F5	1	PLNG	R	SCALR	00F5	1
RDEG	R	SCALR	0029	1	RFA	R	SCALR	00F2	1	PLAT	R	SCALR	00DF	1	PLAT	R	SCALR	00DF	1
RLNG	R	SCALR	0030	1	SIN	R	SPRGG	INTRIN	1	RLAT	R	SCALR	00CA	1	RLAT	R	SCALR	00CA	1
TBDAY	SPRGG	EXTRN			UNPKBY	SPRGG	EXTRN			TC	R	SCALR	00CA	1	TC	R	SCALR	00CA	1

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
2	00066	13	0001E	50	0009E	65	00150
68	00158	70	00191	85	0019A	88	001A0
90	001A3	226	00098	230	0009E	360	0008A
410	001AF	575	0010E	576	00111	578	00126
579	0012C	601	00154	599	0020A	1577	00124
7786	00090						

## LOCAL VARIABLES (1790 WORDS):

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
0000	IBUFOT	00640	MAG	00647	MAG2	00661	IA
00684	IZ	00680	IN	00684	IN	00686	KI
00687	K8	00689	IMEC1	0068A	NEF	0068C	NROUT
0068D	INN	0068F	IN	006C0	NZER8	006C2	KGM88
006C3	KGYR8	006C5	ITAPE	006C6	JTAP	006C8	ELEV
006C9	IREC	006C8	KFA	006CC	IREC	006CE	IFSC
006CF	NRECT	006D1	NREJ	006D2	ICHECK	006D4	DEGRA
006D5	ISORC	006D7	IELC	006D8	IGC	006DA	NFILE
006DB	KGDA	006DD	KGYR	006DE	KGMH	006EO	RLNG

006E1 KVN	006E2 KVE	006E3 K977	006E4 IGR	006E5 KFA	006E6 KRG
006E7 KCVN	006E8 KCVE	006E9 KCDM	006EA MYDC	006EB MT	006EC KETV9
006ED IGRS	006EE DG	006EF KDG	006FO ITEST	006F1 DLAT	006F2 PLAT
006F3 LTKEY	006F4 DLNG	006F5 FLNG	006F6 LGKEY	006F7 8BS9	006F8 FA
006F9 BG	006FA IDEP	006FB J	006FC JKEY	006FD NE	

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

ABS	FLCAT	IFIX	ISL	SIN
-----	-------	------	-----	-----

EXTERNAL SUBPROGRAMS REQUIRED:

BICOR	BUFFER9	EXIT	MCV9L	UNPKBY	F1101
F1102	F1103	F1104	F1106	MDS	MISC
9BCORDEE	9BCOREAD	9BCORIT	9ENOFLE	SINITIAL	9:0DATA
9IOLUSA	9ITOR	9RTOI	9SIN		
			TODAY		
			F1108		
			9ENDI9L		
			9878P		

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	DEC	HEX
CONSTANTS:	WORDS	WORDS
LOCAL VARIABLES:	544	00220
TEMP:	15	0000F
	1790	006FE
	1	00001
	----	-----
TOTAL PROGRAM:	2350	0092E

```

1.  C  PROGRAM G3DCP
2.  C  VERSION OF 1 MAY 75 TO ZERO EVERYTHING IN SIGHT
3.  C  VERSION OF 17 APR 75 FOR XEROX CP-V COMPATIBILITY
4.  C  VERSION OF FEB 75 TO INCREASE PRINT SPACE FOR RH0*Z
5.  C  VERSION 5 FEB 75 TO EASE INTERFACE WITH I/P PREP
6.  C  BUS VERSION OF 22/7/74 TO COMPUTE POTENTIAL FROM BODY
7.  C  OUTPUTS HEIGHT OF SEA (+ DOWN) AND MGAL DIFFERENCE
8.  C  VERSION OF 20 MAY 74 TO INCREASE RESOLUTION OF Z
9.  C  VERSION OF 23 APRIL 73 TO INCREASE RESOLUTION OF PRINTED VALUE OF
10. C      HEIGHT AND WEIGHTEST
11. C  VERSION 29 MARCH 73 TO CHANGE SSW 19 & 20 TO 38 & 39
12. C  VERSION 20 MARCH 1973 TO CALL PLANET FOR RADIUS (LAG)
13. C  3D PROG      FIELD POINTS ARE GIVEN BY K=1, KK. CONTOURS ARE GIVEN  G3DC0010
14. C  BY M=1, MQ. THE VERTICES FOR EACH CONTOUR ARE GIVEN BY I=1, II, II  G3DC0020
15. C  DIFFERENT FOR EACH CONTOUR AND IS GIVEN BY II=E(M), MID(M) IS 1  G3DC0030
16. C  FOR EACH CONTOUR.  G3DC0040
17. C  SENSE SWITCH OPTIONS
18. C      SSW (38)=0 FOR SPHERICAL EARTH
19. C      =1 FOR SPHERICAL MOON
20. C      =2 FOR SPHERICAL MARS
21. C      SSW (39) TO BE RESERVED FOR LATER USE IN SUBR PLANET
22. C  COMMON UIII( 20,10),URH9( 20,10),UZEE( 20,10),UMID( 20,10)  G3DC0050
23. C  COMMON XX(50),YY(50),IA(35),BELP(102),BEL(102),E(102),SIGMA(102)  G3DC0060
24. C  COMMON UU(30),UZU(30),UT(30),GG(102),UZT(30),IMM(30)  G3DC0070
25. C  COMMON III(102),RH0(102),ZEE(102),ZZEE(102),MID(1  G3DC0080
26. C  102),V(102),DEL(102),DEL(102),X(20,20,10),Y(20,20,10)  G3DC0090
27. C  COMMON F(102),PDEL(102),PDEL(102)
28. C  DOUBLE PRECISION RVDR,DRA,DZZ,TT,TX
29. C  INTEGER PD  G3DC0100
30. C  DO 20 I=1,102
31. C  BELP(I)=0.0
32. C  BEL(I)=0.0
33. C  DEL(I)=0.0
34. C  DELP(I)=0.0
35. C  E(I)=0.0
36. C  F(I)=0.0
37. C  GG(I)=0.0
38. C  III(I)=0
39. C  MID(I)=0
40. C  PDEL(I)=0.0
41. C  PDEL(1)=0.0
42. C  RH0(I)=0.0
43. C  SIGMA(I)=0.0
44. C  V(I)=0.0
45. C  ZEE(I)=0.0
46. C  ZZEE(I)=0.0
47. C  20 CONTINUE
48. C  DO 21 I=1,30
49. C  IMM(I)=0
50. C  UT(I)=0.0
51. C  UL(I)=0.0
52. C  UZT(I)=0.0
53. C  UZU(I)=0.0
54. C  21 CONTINUE
55. C  DO 22 I=1,35
56. C  IA(I)=0
57. C  22 CONTINUE
58. C  DO 23 I=1,20
59. C  DO 24 J=1,10

```

```

60.      U111(I,J)=0.0
61.      UMID(I,J)=0.0
62.      URH0(I,J)=0.0
63.      UZEE(I,J)=0.0
64.      24 CONTINUE
65.      23 CONTINUE
66.      D0 25 I=1,20
67.      D9 26 J=1,20
68.      D8 27 K=1,10
69.      X(I,J,K)=0.0
70.      Y(I,J,K)=0.0
71.      27 CONTINUE
72.      26 CONTINUE
73.      25 CONTINUE
74.      D0 28 I=1,50
75.      XX(I)=0.0
76.      YY(I)=0.0
77.      28 CONTINUE
78.      A=0.0
79.      AA=0.0
80.      ALPH1=0.0
81.      ALPH2=0.0
82.      AN0M=0.0
83.      AN0M1=0.0
84.      B=0.0
85.      BB=0.0
86.      BDC=0.0
87.      BETA1=0.0
88.      BETA2=0.0
89.      BG=0.0
90.      C=0.0
91.      CC=0.0
92.      D=0.0
93.      DELT1=0.0
94.      DELT2=0.0
95.      D0G=0.0
96.      D0GG=0.0
97.      D0GGS=0.0
98.      D0GS=0.0
99.      DRA=0.0
100.     DZZ=0.0
101.     EGA=0.0
102.     ELEV=0.0
103.     EGA=0.0
104.     EMM=0.0
105.     FAG=0.0
106.     FELZ=0.0
107.     FX=0.0
108.     FY=0.0
109.     FZ=0.0
110.     GAMM1=0.0
111.     GAMM2=0.0
112.     GM=0.0
113.     I=0
114.     IBEGG=0
115.     IDEP=0
116.     IDIF=0
117.     IFLC=0
118.     IFBC=0

```

120.	IGC=0
121.	II=0
122.	IN=0
123.	INCARD=0
124.	INIT=0
125.	IBUT=0
126.	IRMA=0
127.	ISORC=0
128.	ITAPE=0
129.	ITST=0
130.	JTAPE=0
131.	JTST=0
132.	K=0
133.	KDA=0
134.	KGHM=0
135.	KGM0=0
136.	KGYR=0
137.	KJ=0
138.	KK=0
139.	K977=0
140.	L=0
141.	LDP=0
142.	M=0
143.	MM=0
144.	MN=0
145.	MO=0
146.	MP=0
147.	MRS=0
148.	MUM=0
149.	NGG=0
150.	NG0=0
151.	0BSG=0.0
152.	P=0.0
153.	PAN0M=0.0
154.	PARFEZ=0.0
155.	PB=0.0
156.	PC=0.0
157.	PDENS=0.0
158.	PFELZ=0.0
159.	PIE=0.0
160.	PMASS=0.0
161.	PSFELZ=0.0
162.	PSI=0.0
163.	RA=0.0
164.	RFA=0.0
165.	RHCZ=0.0
166.	RHBZ=0.0
167.	RKM=0.0
168.	RLAT=0.0
169.	RL0NG=0.0
170.	RVAR=0.0
171.	RVDR=0.0
172.	R1=0.0
173.	R2=0.0
174.	R3=0.0
175.	S=0.0
176.	SFELZ=0.0
177.	SHCZ=0.0
178.	SH0Z=0.0
179.	SIGA=0.0

```

180.      SPM=0.0
181.      SPM1=0.0
182.      SPM2=0.0
183.      SS=0.0
184.      STV=0.0
185.      T=0.0
186.      TAU=0.0
187.      TC=0.0
188.      TS=0.0
189.      TT=0.0
190.      TU=0.0
191.      TV=0.0
192.      TX=0.0
193.      U=0.0
194.      W=0.0
195.      Z=0.0
196.      ZT=0.0
197.      ZU=0.0
198.      ZZ=0.0
199.      T=0.0
200.      U=0.0
201.      FAG=0.0
202.      ITAPE=1, JTAPE=2, CALL STAT
203.      OUTPUT 'G3DCF VERSION OF 1 MAY 75'
204.      BD=0
205.      K=0
206.      IOUT=108
207.      INCARD=105
208.      IN=3
209.      ITST=0, JTST=0, INIT=ISW(=2), PIE=.062832
210.      CALL C98RR(FX,FY,RLONG,RLAT,ITST,JTST), JTST=1
211.      KJ=0
212.      CALL GINBT(ITAPE,JTAPE,KJ,KDA,KGM6,KGYR,KGHM,IDI,ISERC,RLAT,
213.      1 RLONG,ELEV,K977,9BSG,IDEF,FAG,BG,TC,IELC,IGC,RFA,IBEGG,IFFC,IA,
214.      2 IFFC)
215.      CALL PLANET(KK,RKM,PMASS,GM,PDENS)
216.      1000 KJ=1
217.      READ(INCARN,1020) RFW,RFD,RFG,J,AUX,VU,VT,GGG,PUN
218.      1020 FORMAT( F8.1,F4.2,F5.1,I1,F2.1,2F12.6,2F2.1)
219.      WRITE(IOUT,1021) RFW,RFD,RFG,J,AUX,VU,VT,GGG,PUN
220.      1021 FORMAT(1H, F8.1,F6.3,F5.1,I5,F4.1,2F12.6,2F4.1)
221.      1001 BD=BD+1
222.      WRITE(IOUT,1003) BD
223.      1003 FORMAT(///,THIS IS BODY 1,12,/)
224.      READ(IN,1002) MQ,UU(BD),UZU(BD),UT(BD),UZT(BD),LBP
225.      1002 FORMAT(12,2(F2.1,F12.6),I1)
226.      WRITE(IOUT,943) MQ,UU(BD),UZU(BD),UT(BD),UZT(BD)
227.      943 FORMAT(14,2(F4.1,F12.6))
228.      MM=MQ+1
229.      IMM(BD)=MM
230.      1025 DO 1064 M=2,MM
231.      READ(IN,1030) UMID(M,BD),URH6(M,BD),UZEE(M,BD),DUM
232.      1030 FORMAT( 12,F10.4,F16.6,F6.2)
233.      URH6(M,BD)=URH6(M,BD)-RFD
234.      WRITE(IOUT,1031)UMID(M,BD),URH6(M,BD),UZEE(M,BD),DUM
235.      1031 FORMAT(1H 12,F10.4,F16.6,F6.2)
236.      II=0
237.      MUM=M-1
238.      IF(DUM)3,4,3
239.      3 IF(M-2)5,4,5

```

G3DC0110

G3DC0120

G3DC0130

G3DC0140

G3DC0150

G3DC0170

G3DC0180

G3DC0190

G3DC0200

G3DC0210

G3DC0220

G3DC0230

G3DC0250

G3DC0260

G3DC0270

G3DC0280

G3DC0290

G3DC0300

G3DC0310

G3DC0320

G3DC0330

G3DC0340

G3DC0350

G3DC0360

G3DC0370

G3DC0380

G3DC0400

G3DC0410

G3DC0430

G3DC0440

G3DC0450

G3DC0460

```

240.      5  DB 6 I=1,U111(MUM,BD)                                G3DC0470
241.      6  X(M,I,BD)=X(MUM,I,BD)                                G3DC0480
242.      G9T8 1050                                                G3DC0490
243.    1041  FORMAT(1H 6F12.5)
244.      4  II=II+1                                                G3DC0510
245.      READ(IN,1040) X(M,II,BD),Y(M,II,BD),LPT                G3DC0520
246.    1040  FORMAT(2F12.5,I1)
247.      IF(LPT)4,4,12                                            G3DC0540
248.      12  U111(M,BD)=II                                          G3DC0550
249.    1050  IF(DUM.EQ.1) G9 T8 1064                                G3DC0560
250.      G9 T8 (1051,1052,1053,1054), 1+ISW(1)+2*ISW(3)          G3DC0570
251.    1051  WRITE(1OUT,1041) (X(M,I,BD),Y(M,I,BD),I=1,II); G9 T8 1064 G3DC0580
252.    1052  DB 1055 I=1,II                                          G3DC0590
253.      CALL C8HRR(XX(I),YY(I),X(M,I,BD),Y(M,I,BD),ITST,JTST)  G3DC0600
254.    1055  X(M,I,BD)=XX(I); Y(M,I,BD)=YY(I)                    G3DC0610
255.      G9 T8 1051                                                G3DC0620
256.    1053  JTST=J                                                G3DC0630
257.      DB 1056 I=1,II                                          G3DC0640
258.      CALL C8HRR(X(M,I,BD),Y(M,I,BD),XX(I),YY(I),ITST,JTST)  G3DC0650
259.    1056  WRITE(1OUT,1041) XX(I),YY(I)                          G3DC0660
260.      G9 T8 1064                                                G3DC0670
261.    1054  DB 1063 I=1,II                                          G3DC0680
262.      CALL C8HRR(X(M,I,BD),Y(M,I,BD),XX(I),YY(I),ITST,JTST)  G3DC0690
263.      WRITE(1OUT,1041) XX(I),YY(I)                              G3DC0700
264.    1063  X(M,I,BD)=XX(I); Y(M,I,BD)=YY(I)                    G3DC0710
265.    1064  CONTINUE                                              G3DC0720
266.      IF(PLA)300,1061,300                                       G3DC0730
267.    300  IF(U)310,320,310                                         G3DC0740
268.    310  MB=1                                                    G3DC0750
269.      UZEE(I,BD)=ZU                                              G3DC0760
270.      E(1)=VU                                                    G3DC0770
271.      V(1)=VU                                                    G3DC0780
272.      F(1)=VU
273.      G9T8 330
274.    320  MB=2                                                    G3DC0790
275.    330  IF(T)340,350,340                                         G3DC0800
276.    340  MP=MM+1                                                  G3DC0810
277.      UZEE(MB,BD)=ZT                                             G3DC0820
278.      V(MP)=VT                                                  G3DC0830
279.      E(MP)=VT                                                  G3DC0840
280.      F(MP)=VT                                                  G3DC0850
281.      G9T8 360
282.    350  MP=MM                                                    G3DC0860
283.    360  NGB=MP-MB+1                                              G3DC0870
284.      MRS=MB+2                                                    G3DC0880
285.      NGG=NGB-2                                                  G3DC0890
286.    1061  IF(LRP) 1001,1001,1060                                  G3DC0900
287.    1060  CONTINUE                                              G3DC0910
288.      BDC=BD                                                    G3DC0920
289.    1070  K=K+1                                                  G3DC0930
290.      BD=0                                                       G3DC0940
291.      JTST=1                                                      G3DC0950
292.      S8M=0.; S8M1=0.                                           G3DC0960
293.      SH8Z=0.; SHCZ=0.                                           G3DC0970
294.      S8M2 = 0.                                                  G3DC0980
295.      WRITE(1OUT,1504)                                           G3DC0990
296.    1504  FORMAT(1H , '*****'//)                                G3DC1000
297.      1*****'////////'//)                                     G3DC1010
298.      WRITE(1OUT,1075)                                           G3DC1020
299.    1075  FORMAT(1H ////95H                                     G3DC1030

```

FIELD POINT COORDINATES

```

300.          1          C&NT&UR DATA          //128H K XG3DC1040
301.          2          Y          Z          VERT. DEPTH DENSITY G3DC1050
302.          3 SIGMA A RH0*Z          DELTA PRIME DELTA /1H ) G3DC1060
303. 1078 F&R&MAT(1H 12,F&8.2,F12.2, F12.2) G3DC1070
304.          IF(1SW(1).EQ.0) G&O T& 17 G3DC1080
305.          IF(1SW(2))1520,1,1520 G3DC1090
306.          1 CALL GIN&ST(IT&PE,JT&PE,KJ,KDA,KGM&,KGYR,KG&M,1DIF,1S&RC,RL&T, G3DC1100
307.          1 RL&NG,ELEV,K977,0BSG,1DEP,F&G,B&G,TC,1ELC,1GC,RFA,1REGG,1FFC,1A, G3DC1110
308.          2 IF&C) G3DC1120
309.          IF(KJ.EQ.9) G&O T& 1520 G3DC1130
310.          CALL CB&RR(FX,FY,RL&NG,RL&T,1T&T,JT&T) G3DC1140
311.          FZ=ELEV/1000. G3DC1150
312.          IF(1SW(3))15,1077,18 G3DC1160
313.          17 RE&D(1NC&RD,1023) FX,FY,FZ,F&G,LDP G3DC1180
314. 1023 F&R&MAT(3(F12.7),F6.2,1) G3DC1190
315.          IF(1SW(3),EQ.1) G&O T& 1072 G3DC1200
316. 1077 WR&TE(1&UT,1078) K,FX,FY,FZ G3DC1210
317.          G&O T& 1062 G3DC1220
318. 1072 JT&T=0 G3DC1230
319.          CALL CB&RR(FX,FY,RL&NG,RL&T,1T&T,JT&T) G3DC1240
320.          18 WR&TE(1&UT,1078) K,RL&T,RL&NG,FZ G3DC1250
321. 1062 BD=BD+1 G3DC1260
322.          WR&TE(1&UT,1003) BD G3DC1270
323.          Z&EE(M&-1)=UZ&EE(M&,BD); RH&Z=0; RH&Z=0 G3DC1280
324.          MM=1MM(BD) G3DC1290
325. 1079 DB1430 M=2,MM G3DC1300
326.          SIGA = 0 G3DC1310
327.          SF&LZ = 0
328.          PS&F&LZ=0
329.          STV=0 G3DC1320
330.          U=UU(BD); ZU=UZU(BD); T=UT(BD); ZT=UZT(BD) G3DC1330
331.          Z&EE(M)=UZ&EE(M,BD); RH&Z(M)=UR&H&Z(M,BD) G3DC1340
332.          111(M)=U111(M,BD); MID(M)=UMID(M,BD) G3DC1350
333.          Z=Z&EE(M)-FZ G3DC1360
334. 1090 ALPH1=X(M,1,BD)-FX G3DC1370
335.          B&TA1=Y(M,1,BD)-FY G3DC1380
336.          R1 = SQ&RTF(ALPH1 ** 2 + B&TA1 ** 2 ) G3DC1390
337.          IF (R1) 1100, 1105, 1100 G3DC1400
338. 1100 GAMM1 = ALPH1 / R1 G3DC1410
339.          DELT1= B&TA1/R1 G3DC1420
340.          1105 IF(AUX)1110,1115,1110 G3DC1430
341. 1110 WR&TE(1&UT,1112) MID(M),111(M),Z&EE(M),RH&Z(M) G3DC1440
342. 1112 F&R&MAT (1H ///12,12H V&RT&ICES=12,9H DEPTH=F7.2,11H DENSITY=F&G G3DC1450
343.          15.2//101H 1 X(1) Y(1) X(1+1) Y(1+1) A G3DC1460
344.          2 B C D PARFEZ //1H ) G3DC1470
345. 1115 11 = 111(M) G3DC1480
346.          DB 1410 1 = 2, 11 G3DC1490
347.          ALPH2=X(M,1,BD)-FX G3DC1500
348.          B&TA2=Y(M,1,BD)-FY G3DC1510
349.          R2 = SQ&RTF (ALPH2 **2 + B&TA2 ** 2 ) G3DC1520
350.          IF (R2) 1120, 1350, 1120 G3DC1530
351. 1120 GAMM2 = ALPH2 / R2 G3DC1540
352.          DELT2 = B&TA2 / R2 G3DC1550
353. 1130 IF (R1) 1140, 1350, 1140 G3DC1560
354. 1140 SS=SQ&RTF((ALPH1-ALPH2)**2 +(B&TA1-B&TA2)**2 ) G3DC1570
355.          E&G&A=(ALPH1-ALPH2)/SS G3DC1580
356.          T&AU=(B&TA1-B&TA2)/SS G3DC1590
357.          P = T&AU * ALPH1 - E&G&A * B&TA1 G3DC1600
358.          IF(ABS&F(P)-.00001)1350,1350,1351 G3DC1610
359. 1351 IF(P)1150,1350,1160 G3DC1620

```

```

360. 1150 S = -1.
361. G0 T0 1170
362. 1160 S = 1.
363. 1170 EMM = BETA1 * ALPH2 - BETA2 * ALPH1
364. 1180 IF (EMM) 1190, 1350, 1200
365. 1190 W = -1.
366. G0 T0 1210
367. 1200 W = 1.
368. 1210 IF (Z) 12101, 12102, 12101
369. 12101 PSI = S * (Z / SQRTF(P**2 + Z**2))
370. 12102 AA = GAMM1 * GAMM2 + DELT1 * DELT2
371. IF (AA) 1225, 1220, 1230
372. 1220 A = W * 1.570796327
373. G0 T0 1240
374. 1225 A = W * (ATANF((SQRTF(1. - AA**2)) / AA) + 3.141592654)
375. G0 T0 1240
376. 1230 A = W * ATANF((SQRTF(1. - AA**2)) / AA)
377. 1240 IF (Z) 12401, 12402, 12401
378. 12402 B = 0
379. C = 0
380. G0 T0 1240
381. 12401 BB = (PSI * (EGA * GAMM1 + TAU * DELT1))
382. IF (BB = 1.) 1260, 1250, 1260
383. 1250 B = 1.570796327
384. G0 T0 1290
385. 1260 IF (BB + 1.) 1280, 1270, 1280
386. 1270 B = -1.570796327
387. G0 T0 1290
388. 1280 B = ATANF(BB / (SQRTF(1. - BB**2)))
389. 1290 CC = (PSI * (EGA * GAMM2 + TAU * DELT2))
390. IF (CC = 1.) 1310, 1300, 1310
391. 1300 C = 1.570796327
392. G0 T0 1340
393. 1310 IF (CC + 1.) 1330, 1320, 1330
394. 1320 C = -1.570796327
395. G0 T0 1340
396. 1330 C = ATANF(CC / (SQRTF(1. - CC**2)))
397. 1340 D = C * B
398. FELZ = A + C
399. BB = EGA * GAMM1 + TAU * DELT1; CC = EGA * GAMM2 + TAU * DELT2
400. IF (1. LE. ABS(BB).OR. 1. LE. ABS(CC)) PB = 1; PC = 1; G0 T0 1341
401. PB = (SQRT(P**2 + (1 - BB**2) * Z**2) * S * P * BB) / SQRT(1 - PB**2)
402. PC = (SQRT(P**2 + (1 - CC**2) * Z**2) * S * P * CC) / SQRT(1 - CC**2)
403. 1341 PFELZ = Z * (D + A) - P * LOG(PB / PC)
404. G0 T0 1360
405. 1350 FELZ = 0
406. PFELZ = C
407. A = 0
408. B = 0
409. C = 0
410. D = 0
411. 1360 IF (AUX) 1370, 1390, 1370
412. 1370 PARFEZ = 6.67 * RH0(M) * FELZ
413. 1380 DEG = ALPH1 + FX
414. DEGS = BETA2 + FY
415. D0GG = ALPH2 + FX
416. D0GGS = BETA2 + FY
417. IRMA = 1
418. WRITE(1907, 1385) IRMA, DEG, DEGS, D0GG, D0GGS, A, B, C, D, PARFEZ
419. WRITE(1907, 9998) SS, TAU, EGA, P

```

G3DC1630  
 G3DC1640  
 G3DC1650  
 G3DC1660  
 G3DC1670  
 G3DC1680  
 G3DC1690  
 G3DC1700  
 G3DC1710  
 G3DC1720  
 G3DC1730  
 G3DC1740  
 G3DC1750  
 G3DC1760  
 G3DC1770  
 G3DC1780  
 G3DC1790  
 G3DC1800  
 G3DC1810  
 G3DC1820  
 G3DC1830  
 G3DC1840  
 G3DC1850  
 G3DC1860  
 G3DC1870  
 G3DC1880  
 G3DC1890  
 G3DC1900  
 G3DC1910  
 G3DC1920  
 G3DC1930  
 G3DC1940  
 G3DC1950  
 G3DC1960  
 G3DC1970  
 G3DC1980  
 G3DC1990  
 G3DC2000  
 G3DC2010

G3DC2020  
 G3DC2030  
 G3DC2040  
 G3DC2050  
 G3DC2060  
 G3DC2070  
 G3DC2080  
 G3DC2090  
 G3DC2100  
 G3DC2110  
 G3DC2120  
 G3DC2130  
 G3DC2140  
 G3DC2150  
 G3DC2160

```

420. 9998 FORMAT(4E18.7) G3DC2170
421. 1385 FORMAT(1H I2,F9.2,F9.2,F9.2,F10.2,F12.7,F12.7,F12.7,F12.7,F12.7,F12.7) G3DC2180
422. 16) G3DC2190
423. 1390 SFELZ=SFELZ+FELZ G3DC2200
424. PSFELZ=PSFELZ+PFELZ
425. SIGA=SIGA+A G3DC2210
426. L=C G3DC2220
427. RA=RKM-ZEE(M); ZZ=RKM-FZ
428. DRA=DBLE(RA)
429. DZZ=DBLE(ZZ)
430. RVAR=R1 G3DC2240
431. 1398 TS=1-Z/SQRTF(Z**2+RVAR**2) G3DC2250
432. RVDR=DBLE(RVAR)
433. TT=DGR1((DRA*DSIN(RVDR/DRA))**2+(DZZ*DRA*DCBS(RVDR/DRA))**2)
434. TX=DRA*(2*DRA+TT-(DZZ**2*DRA**2)/TT)/(2*DZZ**2)
435. TU=SGL(TX)-TS
436. L=L+1 G3DC2290
437. G8 TR(1392,1393,1394,1395,1396),L G3DC2300
438. 1392 TV=9*TL G3DC2310
439. RVAR=R2 G3DC2320
440. G8 TR 1398 G3DC2330
441. 1393 TV=TV+9*TL G3DC2340
442. RVAR=R1*R2*SIN(A)/((R1+R2)*SIN(A/2)) G3DC2350
443. R3=RVAR G3DC2360
444. G8 TR 1398 G3DC2370
445. 1394 TV=TV+22*TL G3DC2380
446. RVAR=R1*R3*SIN(A/2)/((R1+R3)*SIN(A/4)) G3DC2390
447. G8 TR 1398 G3DC2400
448. 1395 TV=TV+28*TL G3DC2410
449. RVAR=R2*R3*SIN(A/2)/((R2+R3)*SIN(A/4)) G3DC2420
450. G8 TR 1398 G3DC2430
451. 1396 TV=(TV+28*TL)*(A/96) G3DC2440
452. STV=STV+TV G3DC2450
453. 1400 ALPH1=ALPH2 G3DC2460
454. BETA1=BETA2 G3DC2470
455. GAMM1=GAMM2 G3DC2480
456. DELT1=DELT2 G3DC2490
457. R1=R2 G3DC2500
458. 1410 CONTINUE G3DC2510
459. 1420 IF(SIGA)201,202,203 G3DC2520
460. 201 IF(SIGA+.00001)204,205,205 G3DC2530
461. 205 SFELZ=SFELZ-SIGA G3DC2540
462. G8TR202 G3DC2550
463. 203 IF(SIGA+.00001)205,205,206 G3DC2560
464. 204 IF(SIGA+6.2831754)207,207,2021 G3DC2570
465. 2021 IF(SIGA+3.1416027)202,2023,2022 G3DC2580
466. 2022 IF(SIGA+3.1415827)2023,2023,202 G3DC2590
467. 2023 SFELZ=SFELZ-SIGA-3.1415927 G3DC2600
468. G8TR 202 G3DC2610
469. 207 SFELZ=SFELZ-SIGA-6.2831854 G3DC2620
470. G8TR202 G3DC2630
471. 206 IF(SIGA+6.2831754)2024,208,208 G3DC2640
472. 2024 IF(SIGA-3.1415827)202,2025,2026 G3DC2650
473. 2026 IF(SIGA-3.1416027)2025,2025,202 G3DC2660
474. 2025 SFELZ=SFELZ-SIGA+3.1415927 G3DC2670
475. G8TR 202 G3DC2680
476. 208 SFELZ=SFELZ-SIGA+6.2831854 G3DC2690
477. 202 V(M)=6.67*RH0(M)*(SFELZ+STV) G3DC2700
478. E(M)=6.67*RH0(M)*SFELZ G3DC2710
479. F(M)=6.67*RH0(M)*PSFELZ

```

```

480.      SIGMA(M)=SIGA
481.      1430 CCONTINUE
482.      IF(U)1600,1610,1600
483.      1600 M8=1
484.      MID(1)=0
485.      III(1)=1
486.      ZEE(1)=ZU
487.      RH0(1)=RH0(2)
488.      SIGMA(1)=0.
489.      V(1)=VU
490.      E(1)=VU
491.      F(1)=VU
492.      GO TO 1620
493.      1610 M8=2
494.      1620 IF(T)1630,1632,1630
495.      1630 MP=MM+1
496.      MID(MP)=MID(MM)+1
497.      III(MP)=1
498.      ZEE(MP)=ZT
499.      RH0(MP)=RH0(MM)
500.      SIGMA(MP)=0.
501.      E(MP)=VT
502.      V(MP)=VT
503.      F(MP)=VT
504.      GO TO 1432
505.      1632 MP=MM
506.      1432 DEL(M8)=0.
507.      DELP(M8)=0.
508.      DELP(M8+1)=0.
509.      DEL(M8)=0.
510.      ANBM=0
511.      BEL(M8)=0.
512.      BELP(M8)=0.
513.      BELP(M8+1)=0.
514.      BEL(MP)=0.
515.      ANBM1=0
516.      PDEL(M8)=0.
517.      PDELP(M8)=0.
518.      PDELP(M8+1)=0.
519.      PDEL(MP)=0.
520.      PANBM=0
521.      MN=MP-2
522.      144C DO 1450 M=MB,MN
523.      DEL(M+1) = (V(M)*((ZEE(M)-ZEE(M+1))/(ZEE(M)-ZEE(M+2))))*
524.      1(3.0*ZEE(M+2)-2.*ZEE(M)-ZEE(M+1))+V(M+1)*((ZEE(M)-ZEE(M+1))/(ZEE(M+1)-ZEE(M+2))))*(3.*ZEE(M+2)-2.*ZEE(M+1)-ZEE(M+2)))/6.0
525.      2E(M+1))/(ZEE(M+1)-ZEE(M+2))))*(3.*ZEE(M+2)-2.*ZEE(M+1)-ZEE(M+2)))/6.0
526.      3ZEE(M)+V(M+2)*((ZEE(M)-ZEE(M+1))*3)/(ZEE(M+1)-ZEE(M+2)))/6.0
527.      4(M+2))*((ZEE(M)-ZEE(M+2))))/6.0
528.      BEL(M+1) = (E(M)*((ZEE(M)-ZEE(M+1))/(ZEE(M)-ZEE(M+2))))*
529.      1(3.0*ZEE(M+2)-2.*ZEE(M)-ZEE(M+1))+E(M+1)*((ZEE(M)-ZEE(M+1))/(ZEE(M+1)-ZEE(M+2))))*(3.*ZEE(M+2)-2.*ZEE(M+1)-ZEE(M+2)))/6.0
530.      2E(M+1))/(ZEE(M+1)-ZEE(M+2))))*(3.*ZEE(M+2)-2.*ZEE(M+1)-ZEE(M+2)))/6.0
531.      3ZEE(M)+E(M+2)*((ZEE(M)-ZEE(M+1))*3)/(ZEE(M+1)-ZEE(M+2)))/6.0
532.      4(M+2))*((ZEE(M)-ZEE(M+2))))/6.0
533.      PDEL(M+1) = (F(M)*((ZEE(M)-ZEE(M+1))/(ZEE(M)-ZEE(M+2))))*
534.      1(3.0*ZEE(M+2)-2.*ZEE(M)-ZEE(M+1))+F(M+1)*((ZEE(M)-ZEE(M+1))/(ZEE(M+1)-ZEE(M+2))))*(3.*ZEE(M+2)-2.*ZEE(M+1)-ZEE(M+2)))/6.0
535.      2E(M+1))/(ZEE(M+1)-ZEE(M+2))))*(3.*ZEE(M+2)-2.*ZEE(M+1)-ZEE(M+2)))/6.0
536.      3ZEE(M)+F(M+2)*((ZEE(M)-ZEE(M+1))*3)/(ZEE(M+1)-ZEE(M+2)))/6.0
537.      4(M+2))*((ZEE(M)-ZEE(M+2))))/6.0
538.      DELP(M+2) = (V(M)*((ZEE(M+1)-ZEE(M+2))*3)/(ZEE(M+1)-ZEE(M+2))))*
539.      1 - ZEE(M+1))*((ZEE(M)-ZEE(M+2)))+V(M+1)*((ZEE(M+1)-ZEE(M+2))))*

```

G3DC2720  
G3DC2730  
G3DC2740  
G3DC2750  
G3DC2760  
G3DC2770  
G3DC2780  
G3DC2790  
G3DC2800  
G3DC2810  
G3DC2820  
  
G3DC2830  
G3DC2840  
G3DC2850  
G3DC2860  
G3DC2870  
G3DC2880  
G3DC2890  
G3DC2900  
G3DC2910  
G3DC2920  
G3DC2930  
  
G3DC2940  
G3DC2950  
G3DC2960  
G3DC2970  
G3DC2980  
G3DC2990  
G3DC3000  
G3DC3010  
G3DC3020  
G3DC3030  
G3DC3040  
G3DC3050

G3DC3060  
G3DC3070  
G3DC3080  
G3DC3090  
G3DC3100  
G3DC3110  
G3DC3120  
G3DC3130  
G3DC3140  
G3DC3150  
G3DC3160  
G3DC3170  
G3DC3180  
G3DC3190

```

540. 2(M+2) / (ZEE(M) - ZEE(M+1)) * (ZEE(M+2) + 2. * ZEE(M+1) - 3. * ZEE(M)) / 6.0 G3DC3200
541. 3(M) + V(M+2) * ((ZEE(M+1) - ZEE(M+2)) / (ZEE(M) - ZEE(M+2))) * (ZEE(M+2) - ZEE(M)) / 6.0 G3DC3210
542. 4E(M+1) + 2. * ZEE(M+2) - 3. * ZEE(M)) / 6.0 G3DC3220
543. BELP(M+2) = (E(M) * ((ZEE(M+1) - ZEE(M+2)) * 3) / ((ZEE(M) - ZEE(M+2)) * 3)) / ((ZEE(M) - ZEE(M+2)) * 3) G3DC3230
544. 1) - ZEE(M+1)) * (ZEE(M) - ZEE(M+2)) + E(M+1) * ((ZEE(M+1) - ZEE(M+2)) * 3) / ((ZEE(M) - ZEE(M+2)) * 3) G3DC3240
545. 2(M+2) / (ZEE(M) - ZEE(M+1)) * (ZEE(M+2) + 2. * ZEE(M+1) - 3. * ZEE(M)) / 6.0 G3DC3250
546. 3(M) + E(M+2) * ((ZEE(M+1) - ZEE(M+2)) / (ZEE(M) - ZEE(M+2))) * (ZEE(M+2) - ZEE(M)) / 6.0 G3DC3260
547. 4E(M+1) + 2. * ZEE(M+2) - 3. * ZEE(M)) / 6.0 G3DC3270
548. PDEL(M+2) = (F(M) * ((ZEE(M+1) - ZEE(M+2)) * 3) / ((ZEE(M) - ZEE(M+2)) * 3)) / ((ZEE(M) - ZEE(M+2)) * 3) G3DC3280
549. 1) - ZEE(M+1)) * (ZEE(M) - ZEE(M+2)) + F(M+1) * ((ZEE(M+1) - ZEE(M+2)) * 3) / ((ZEE(M) - ZEE(M+2)) * 3) G3DC3290
550. 2(M+2) / (ZEE(M) - ZEE(M+1)) * (ZEE(M+2) + 2. * ZEE(M+1) - 3. * ZEE(M)) / 6.0 G3DC3300
551. 3(M) + F(M+2) * ((ZEE(M+1) - ZEE(M+2)) / (ZEE(M) - ZEE(M+2))) * (ZEE(M+2) - ZEE(M)) / 6.0 G3DC3310
552. 4E(M+1) + 2. * ZEE(M+2) - 3. * ZEE(M)) / 6.0 G3DC3320
553. 1450 CONTINUE G3DC3280
554. ANBM = 0.5 * (DEL(M+1) + DELP(M)) G3DC3290
555. ANBM1 = 0.5 * (BEL(M+1) + BELP(M)) G3DC3300
556. PANBM = 0.5 * (PDEL(M+1) + PDELP(M)) G3DC3310
557. DB1460M = M, MP G3DC3320
558. ANBM = ANBM + 0.5 * (DEL(M) + DELP(M)) G3DC3330
559. ANBM1 = ANBM1 + 0.5 * (BEL(M) + BELP(M)) G3DC3340
560. PANBM = PANBM + 0.5 * (PDEL(M) + PDELP(M)) G3DC3350
561. GG(M) = ANBM - 0.5 * DELP(M) G3DC3360
562. 1460 CONTINUE G3DC3370
563. IF(K-1) 1451, 1451, 1454 G3DC3380
564. 1451 IF(LFG) 1453, 1454, 1454 G3DC3390
565. 1453 RFG = ANBM G3DC3400
566. 1454 GG(M) = 0.0 G3DC3410
567. GG(M+1) = 0.0 G3DC3420
568. GG(MP) = GG(MP) + 0.5 * DELP(MP) G3DC3430
569. DB 1471 M = M, MP G3DC3440
570. ZZEE(M) = ZEE(M) G3DC3450
571. RH0Z = RH0Z + RH0(M) * (ZZEE(M) - ZZEE(M-1)) * SIGMA(M) G3DC3460
572. RHCZ = RHCZ + (ZZEE(M) - ZZEE(M-1)) * SIGMA(M) G3DC3470
573. IF(M-M0) 1532, 1532, 1533 G3DC3480
574. 1532 RH0Z = 0; RHCZ = 0 G3DC3490
575. 1533 CONTINUE G3DC3500
576. WRITE(18UT, 1470) MID(M), III(M), ZEE(M), RH0(M), SIGMA(M) G3DC3510
577. 1 (M), RH0(M) * SIGMA(M) * (ZZEE(M) - ZZEE(M-1)) / PIE, V(M), DELP(M), DEL(M) G3DC3520
578. 1471 CONTINUE G3DC3530
579. 1470 FORMAT (42H 12, 15, E1 G3DC3540
580. 10.3, F8.2, F12.7, F12.5, F12.6, 2E12.4) G3DC3550
581. PANBM = PANBM / 980. G3DC3560
582. C PANBM IS THE POTENTIAL PANBM/980 IS THE HEIGHT IN METERS G3DC3570
583. WRITE(18UT, 1480) ANBM, ANBM1, RFG, ANBM1, RFG, FAG, FAG, ANBM1, RH0Z / PIE - RFW, G3DC3580
584. 1 PANBM = 3086, PANBM, RHCZ / PIE - RFW G3DC3590
585. S0M = S0M + ANBM G3DC3600
586. S0M1 = S0M1 + ANBM1 G3DC3610
587. S0M2 = PANBM + S0M2 G3DC3620
588. SH0Z = SH0Z + RH0Z; SHCZ = SHCZ + RHCZ G3DC3630
589. 1501 IF(BDC-BD) 1502, 1502, 1062 G3DC3640
590. 1502 WRITE(18UT, 1503) S0M, S0M - RFG, S0M - RFG - FAG, FAG, S0M1, G3DC3650
591. 1 SH0Z / PIE - RFW, S0M2, 3086, S0M2, SHCZ / PIE - RFW G3DC3660
592. 1503 FORMAT(///, ' ***** THESE ARE THE FINAL SUMS *****', //, G3DC3670
593. 1, 'CURVED ANOMALY = 'E12.4, ' REFER ANOMALY = 'E12.4, G3DC3680
594. 2, 'RESID ANOMALY = 'E12.4, ' OBSER ANOMALY = 'E12.4, //, G3DC3690
595. 3, 'FLAT ANOMALY = 'E12.4, T94, 'WEIGHT = 'E12.7, //, G3DC3700
596. 4, 'FLAT SEASURF = 'E12.6, ' HEIGHT METERS = 'E14.6, T91, 'WEIGHTEST' G3DC3710
597. 5, 'E12.4) G3DC3720
598. 1480 FORMAT(' '///, 'CURVED ANOMALY = 'E12.4, ' REFER ANOMALY = 'E12.4, G3DC3730
599. 1, 'RESID ANOMALY = 'E12.4, ' OBSER ANOMALY = 'E12.4, //, 'FLAT ' G3DC3740

```

600.	2	ANNNALY=	'E12.4,T94,WEIGHT=	'E12.4,/'	
601.	3	FLAT SEASURF=	'E12.6,1	HEIGHT METERS=	'E14.6,T91,WEIGHTEST'
602.	4	*	'E12.4)		
603.		IF(PUN)	390,7171,390		G3DC3670
604.	390	WRITE(OUTPUT,400)	(V(M),M=M6,MP)		G3DC3680
605.	370	FORMAT(214)			G3DC3690
606.	380	FORMAT(6E12.6)			G3DC3700
607.	400	FORMAT(6E12.6)			G3DC3710
608.	7171	IF(GG)	14701,1500,14701		G3DC3720
609.	14701	WRITE(OUTPUT,14702)	(GG(M),M=M6,MP)		G3DC3730
610.		IF(PUN)	1473,1500,1473		G3DC3740
611.	1473	WRITE(OUTPUT,400)	(GG(M),M=MRS,MP)		G3DC3750
612.	14702	FORMAT(1H 9E12.4)			G3DC3760
613.	1500	IF(LDP)	1070,1070,1520		G3DC3770
614.	1510	IF(J)	1000,1520,1000		G3DC3780
615.	1520	CONTINUE			G3DC3790
616.		END			G3DC3800

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	0000F	1	AA	R	SCALR	0000F	1	ABS	R	SPR8G	INTRIN	1
ABSF	R	SPR8G	00010	1	ALPH1	R	SCALR	00010	1	ALPH2	R	SCALR	00011	1
ABSM	R	SCALR	00012	1	ANGV1	R	SCALR	00013	1	ATANE	R	SPR8G	EXTEN	1
AUX	R	SCALR	00022	1	B	R	SCALR	00014	1	BB	R	SCALR	00015	102
BD	R	SCALR	00024	1	BDC	R	SCALR	00016	1	BEL	R	ARRAY	00018	1
BELP	R	ARRAY	00027	102	BETA1	R	SCALR	00017	1	BETA2	R	SCALR	00019	1
BG	R	SCALR	00019	1	C	R	SCALR	0001A	1	CC	R	SCALR	0001B	1
CBRR	R	SPR8G	EXTEN	1	D	R	SCALR	0001C	1	DBLE	R	SPR8G	INTRIN	102
CCS	R	SPR8G	INTRIN	1	DEL	R	ARRAY	0001E	102	DELP	R	ARRAY	0001F	102
DLT1	R	SCALR	0001D	1	DEL2	R	SCALR	0001E	1	D8C	R	SCALR	00020	1
DGG	R	SCALR	0002C	1	D8GS	R	SCALR	00021	1	D8S	R	SCALR	00022	1
DRA	R	SCALR	00022	2	DSIN	R	SPR8G	INTRIN	1	DSORT	R	SPR8G	INTRIN	102
DUM	R	SCALR	00023	1	DZZ	R	SCALR	00024	1	E	R	ARRAY	00025	1
F	R	SCALR	00023	102	ELEV	R	SCALR	00026	1	EMM	R	SCALR	00027	1
FX	R	ARRAY	02XAP	1	FAG	R	SCALR	00026	1	FELZ	R	SCALR	00027	1
GAMH1	R	SCALR	00028	1	FV	R	SCALR	00029	1	GG	R	ARRAY	0002A	102
GGG	R	SCALR	00028	1	GAM2	R	SCALR	0002C	1	GM	R	ARRAY	0002D	1
I	R	SCALR	00085	1	GINOT	R	SPR8G	EXTEN	1	IBEGG	R	SCALR	0002E	1
IEP	R	SCALR	0000F	1	IA	R	ARRAY	00034	35	IELC	R	SCALR	00031	1
IFBC	R	SCALR	0002F	1	IDIF	R	SCALR	00030	1	IGC	R	SCALR	00034	30
II	R	SCALR	00032	1	IFFC	R	SCALR	00033	1	IMM	R	ARRAY	00038	1
IN	R	SCALR	00035	1	III	R	ARRAY	0003B	102	INIT	R	SCALR	00038	1
IBUT	R	SCALR	00036	1	INCARD	R	SCALR	00037	1	ISORC	R	SCALR	0003B	1
ISW	R	SCALR	00039	1	IRMA	R	SCALR	0003A	1	ITST	R	SCALR	0003D	1
J	R	SPR8G	EXTEN	1	ITAPE	R	SCALR	0003C	1	JTST	R	SCALR	0003F	1
K	R	SCALR	0000C	1	JTAPE	R	SCALR	0003E	1	KJ	R	SCALR	00041	1
KGM0	R	SCALR	0000D	1	KDYR	R	SCALR	00040	1	KJM	R	SCALR	00044	1
KK	R	SCALR	00042	1	KDYR	R	SCALR	00043	1	L	R	SCALR	00047	1
LBP	R	SCALR	00045	1	K977	R	SCALR	00046	1	LBG	R	SPR8G	INTRIN	102
LPT	R	SCALR	00048	1	LDP	R	SCALR	00048	1	MID	R	ARRAY	0004C	1
MH	R	SCALR	0008A	1	M	R	SCALR	00049	1	M8	R	SCALR	0004E	1
MP	R	SCALR	0004A	1	MN	R	SCALR	0004B	1	MRS	R	SCALR	00051	1
MUM	R	SCALR	0004F	1	MC	R	SCALR	00048	1	NG8	R	SCALR	00054	1
MSG	R	SCALR	0004F	1	NG	R	SCALR	00050	1	PANM	R	SCALR	00054	1
MSG	R	SCALR	00052	1	P	R	SCALR	00053	1	PC	R	SCALR	00057	1
PARFEZ	R	SCALR	00052	1	PB	R	SCALR	00053	1	PDENS	R	SCALR	00058	1
PEL	R	ARRAY	02911	102	PBPLP	R	SCALR	00056	1	PLANET	R	SPR8G	EXTEN	1
PELZ	R	SCALR	00059	1	PIE	R	SCALR	0005A	1	PSI	R	SCALR	0005D	1
PMASS	R	SCALR	0005B	1	PIEFELZ	R	SCALR	0005E	1	RFA	R	SCALR	0005F	1
PUN	R	SCALR	0005B	1	RA	R	SCALR	0005E	1	RFA	R	SCALR	0007F	1
RED	R	SCALR	0008C	1	RFG	R	SCALR	00081	1	RFB	R	SCALR	00061	1
RHCZ	R	SCALR	0008C	1	RH9	R	SCALR	00081	102	RH02	R	SCALR	00064	1
RKM	R	SCALR	00062	1	RLAT	R	SCALR	00063	1	RH02	R	SCALR	00066	1
RVAR	R	SCALR	0006E	1	RVDH	R	SCALR	00063	2	R1	R	SCALR	00069	1
R2	R	SCALR	00067	1	R3	R	SCALR	00068	1	S	R	SCALR	0006C	1
SEELZ	R	SCALR	0006A	1	SHCZ	R	SCALR	0006B	1	SH02	R	SCALR	0006C	1
SIGA	R	SCALR	0006D	1	SIGA	R	SCALR	0006B	1	SIN	R	SCALR	0006F	1
SIGL	R	SCALR	0006D	1	SIGA	R	SCALR	0006B	1	SIN	R	SCALR	0006F	1
SS	R	SCALR	00073	1	SSM	R	SCALR	00069	102	SORT	R	SCALR	0007F	1
SS	R	SCALR	00073	1	SSM	R	SCALR	00069	1	SORT	R	SCALR	0007F	1
T	R	SCALR	00073	1	STAT	R	SPR8G	EXTEN	1	STV	R	SCALR	00075	1
TS	R	SCALR	00073	1	TAU	R	SCALR	00074	1	TC	R	SCALR	00075	1
TV	R	SCALR	00078	1	TT	R	SCALR	00074	2	TU	R	SCALR	00077	1
UII1	R	SCALR	00078	200	TX	R	SCALR	00078	2	U	R	SCALR	00079	1
UT	R	ARRAY	00078	30	UMID	R	SCALR	00058	200	URH9	R	ARRAY	000C8	200
UZY	R	ARRAY	00078	30	UU	R	SCALR	0005F	30	UZEE	R	ARRAY	00190	200
					UZU	R	SCALR	0005F	30				00339	102

[illegible]

00000	RVDR	00002	DRA	00004	DZZ	00006	TT	00008	TX	0000A	BD
00001	I	00003	JNFM	00005	KANFM	00007	A	00009	AA	00011	ALPH1
00002	ALPH2	00004	BETA2	00006	BG	00008	B	00010	AB	00012	BDC
00003	BETA1	00005	DELTA1	00007	DMG	00009	C	00011	BB	00013	D
00004	DELTA2	00006	ELEV	00008	EMM	00010	D	00012	CC	00014	D6GS
00005	EGA	00007	FZ	00009	GAMM1	00011	DGG	00013	CC	00015	D6GS
00006	FV	00008	IDIF	00010	GAMM2	00012	FAG	00014	CC	00016	D6GS
00007	IDEP	00009	IDIF	00011	IELC	00013	FAG	00015	CC	00017	FX
00008	ISBC	00010	IN	00012	INCRD	00014	FBNC	00016	CC	00018	IBEGG
00009	ISBC	00011	IN	00013	INCRD	00015	INIT	00017	CC	00019	IGC
00010	KGM	00012	KGW	00014	ITST	00016	JTPE	00018	CC	00020	IRMA
00011	L	00013	KGM	00015	KGYR	00017	KJ	00019	CC	00021	JTST
00012	L	00014	LDP	00016	M	00018	KJ	00020	CC	00022	K977
00013	M	00015	MRS	00017	M	00019	MM	00021	CC	00023	M8
00014	N	00016	N	00018	MUM	00020	MM	00022	CC	00024	MB
00015	N	00017	N	00019	NUM	00021	MM	00023	CC	00025	MB
00016	N	00018	N	00020	PARLEZ	00022	MM	00024	CC	00026	MB
00017	N	00019	N	00021	PNASS	00023	MM	00025	CC	00027	MB
00018	N	00020	N	00022	RH9Z	00024	MM	00026	CC	00028	MB
00019	N	00021	N	00023	RH9Z	00025	MM	00027	CC	00029	MB
00020	N	00022	N	00024	R2	00026	MM	00028	CC	00030	MB
00021	N	00023	N	00025	SIGA	00027	MM	00029	CC	00031	MB
00022	N	00024	N	00026	SIGA	00028	MM	00030	CC	00032	MB
00023	N	00025	N	00027	SIGA	00029	MM	00031	CC	00033	MB
00024	N	00026	N	00028	SIGA	00030	MM	00032	CC	00034	MB
00025	N	00027	N	00029	SIGA	00031	MM	00033	CC	00035	MB
00026	N	00028	N	00030	SIGA	00032	MM	00034	CC	00036	MB
00027	N	00029	N	00031	SIGA	00033	MM	00035	CC	00037	MB
00028	N	00030	N	00032	SIGA	00034	MM	00036	CC	00038	MB
00029	N	00031	N	00033	SIGA	00035	MM	00037	CC	00039	MB
00030	N	00032	N	00034	SIGA	00036	MM	00038	CC	00040	MB
00031	N	00033	N	00035	SIGA	00037	MM	00039	CC	00041	MB
00032	N	00034	N	00036	SIGA	00038	MM	00040	CC	00042	MB
00033	N	00035	N	00037	SIGA	00039	MM	00041	CC	00043	MB
00034	N	00036	N	00038	SIGA	00040	MM	00042	CC	00044	MB
00035	N	00037	N	00039	SIGA	00041	MM	00043	CC	00045	MB
00036	N	00038	N	00040	SIGA	00042	MM	00044	CC	00046	MB
00037	N	00039	N	00041	SIGA	00043	MM	00045	CC	00047	MB
00038	N	00040	N	00042	SIGA	00044	MM	00046	CC	00048	MB
00039	N	00041	N	00043	SIGA	00045	MM	00047	CC	00049	MB
00040	N	00042	N	00044	SIGA	00046	MM				

0007C ZU	0007E ZZ	0007F RFW	00080 RFD	00081 RFG	00082 AUX
00083 VU	00084 VT	00085 GGG	00086 PUN	00087 MQ	00088 LBP
00089 DUM	0008A LPT				

## BLANK COMMON (10717 WORDS):

00000 U111	00120 UZEE	00258 UMID	00320 XX	00352 YY
00384 IA	00400 BEL	00473 E	004D9 SIGMA	0053F UU
00550 UZU	00599 GG	005FF UZT	0061D IMM	0063B I11
006A1 RH8	0076D ZEE	007D3 MID	00839 V	0089F DEL
00905 DELP	0190B Y	028AB F	02911 PDEL	02977 PDELP

## INTRINSIC SUBPROGRAMS USED:

ABS	ABSF	DALE	DC8S	DSIN	DSORT	LEG	SIN
SNGL	SGRT						

## EXTERNAL SUBPROGRAMS REQUIRED:

ATANF	C89RR	GINGT	ISW	PLANET	SQRTF	STAT	F:101
F:102	F:103	F:104	F:105	F:106	F:108	MID8	M:8C
9AL9G	9BCDREAD	9BCDWRIT	9DC8S	9DSIN	9DSORT	9DT8R	9END18L
9INITIAL	918DATA	91T8R	9PRINT	9RT8I	9SIN	9SQRT	9STOP

## HIGHEST ERROR SEVERITY: C (NO ERRORS)

DEC	HEX
WORDS	WORDS

GENERATED CODE:	2885	00845
CONSTANTS:	38	00026
LOCAL VARIABLES:	139	00088
TEMPS:	12	0000C

TOTAL PROGRAM:	3074	00002
----------------	------	-------

(PLUS BLANK COMMON)

```

1.  C    PROGRAM G3DC PREP
2.  C    VERSION 5 FEB 75 TO USE REFERENCE DENSITY
3.  C    VERSION OF 23 MAY 74 TO INCREASE RESOLUTION OF Z
4.  C    VERSION 1 MAR 74 TO RENUMBER LAMINAE
5.  C    PROGRAM TO PREPARE I/P TO G3D
6.      DIMENSION ICARD(80),NSL(10)
7.      DIMENSION REFD(10)
8.      OUTPUT 'G3DCPREP VERSION 5 FEB 75'
9.      LBP=1
10.     ITAPE=0
11.     JTAPE=7
12.     IIN=105
13.     100 READ(IIN,1004) NUMB0D
14.         DO 105 I=1,NUMB0D
15.         READ(IIN,1006) REFD(I)
16.     105 CONTINUE
17.         DO 110 I=1,NUMB0D
18.         READ(13,1004) NSLCNT
19.         NSL(I)=NSLCNT
20.     110 CONTINUE
21.         DO 500 I=1,NUMB0D
22.         NCNTP=0
23.         ITAPE=ITAPE+1
24.         IF(I.EQ.NUMB0D) WRITE(JTAPE,1005) NSL(I),LBP , GO TO 130
25.         WRITE(JTAPE,1004) NSL(I)
26.     130 CONTINUE
27.     170 READ(ITAPE,1002,END=400) NCNT,RH0,Z
28.         NCNTP=NCNTP+1
29.         RH0=RH0-REFD(I)
30.         WRITE(JTAPE,1002) NCNTP,RH0,Z
31.     180 READ(ITAPE,1003) X,Y,LSLPT
32.         WRITE(JTAPE,1003) X,Y,LSLPT
33.         IF(LSLPT.NE.1) GO TO 180
34.         GO TO 170
35.     400 CONTINUE
36.     500 CONTINUE
37.     STOP
38.  C    *****
39.  C    FORMATS
40.  C    *****
41.     1001 FORMAT(80A1)
42.     1002 FORMAT(I2,F10.4,F16.6,F6.3)
43.     1003 FORMAT(2F12.5,I1)
44.     1004 FORMAT (I2)
45.     1005 FORMAT(I2,28X,I1)
46.     1006 FORMAT(F10.0)
47.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
I	I	SCALR	00069 V	1	ICARD	I	ARRAY	00000 V	80	IIN	I	SCALR	00067 V	1
ITAPE	I	SCALR	00065 V	1	JTAPE	I	SCALR	00066 V	1	LBP	I	SCALR	00064 V	1
LSLPT	I	SCALR	00071 V	1	NCNT	I	SCALR	0006C V	1	NCNTP	I	SCALR	00068 V	1
NSL	I	ARRAY	00050 V	10	NSLCNT	I	SCALR	0006A V	1	NUMB8D	I	SCALR	00068 V	1
REFC	R	ARRAY	0005A V	10	RH8	R	SCALR	0006D V	1	X	R	SCALR	0006F V	1
Y	R	SCALR	00070 V	1	Z	R	SCALR	0006E V	1					

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
100	00015	110	00031	170	0004B
400	00072	1001	00079	1003	00083
1005	00089	1006	0008D	1004	00087

## LOCAL VARIABLES (114 WORDS):

00000	ICARD	00050	NSL
00067	IIN	00068	NUMB8D
0006D	RH8	0006E	Z

## BLANK COMMON (0 WORDS)

## EXTERNAL SUBPROGRAMS REQUIRED:

F:101	F:102	F:103	F:104	F:105	F:106	F:108	M:08
M:0C	9BCCRDEE	9BCCREAD	9BCCWRIT	9ENDI0L	SINITIAL	9IBDATA	SPRINT
9ST0P							

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
146	00092
0	00000
114	00072
0	00000
260	00104

GENERATED CODE: 146  
 CONSTANTS: 0  
 LOCAL VARIABLES: 114  
 TEMPS: 0  
 TOTAL PROGRAM: 260

```

1.  C  PROGRAM HIG
2.  C  FOR INITIAL CONVERSION OF H.I.G. GRAVITY DATA TO WH91 GSUM
3.  C  INITIAL VERSION BASED ON PROGRAM DMA
4.  C  INITIAL VERSION 1 DECEMBER 1973
5.  DIMENSION IA(35),NAME(80)
6.  DIMENSION IZ(9), IW(35)
7.  DATA IS/' ' S'/
8.  DATA IWE/' ' W'/
9.  ITAPE=1
10.  UTAPE=2
11.  IIN=105
12.  IOUT=108
13.  OUTPUT 'PROGRAM HIG VERSION 2 DEC 73'
14.  CALL STAT
15.  J=ISW(-2)
16.  DEGRA=1.745329E-2
17.  RADEG=57.29578
18.  KK=0
19.  KI=1
20.  KG=-2
21.  C  DENS=2.67-1.03
22.  DENS=1.64
23.  NREC=0
24.  ELEV=0.
25.  READ(IIN,1001) ISORC
26.  1001 FORMAT(I5)
27.  OUTPUT ISORC
28.  CALL GINOT(ITAPE,UTAPE,KK,KGDA,KGM0,
29.  1 KGYR,KGMM,JDIF,ISORC,RLAT,RLONG,ELEV,K977,OBSSG,
30.  2 IDEP,FA,BG,TC,CELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
31.  IFFC=17
32.  IGC=0
33.  IRECC=0
34.  IFBC=0
35.  READ(ITAPE,1002) IHSC,ISIGFA,ISIGBG
36.  1002 FORMAT(10X,I4,I2,I2)
37.  READ(ITAPE,1003)(NAME(I),I=1,80)
38.  1003 FORMAT(80A1)
39.  50 CONTINUE
40.  READ(ITAPE,1004,END=999) IL9C,ITR,ISER,ISTA,ILATD,DLATM,NORS,
41.  1 IL9NGD,DL9NGM,NE9RW,IELEV,NG1,NG2,IELEVKEY,IYR,IAPP,
42.  2 IELEVC,IELEVT,IFA
43.  1004 FORMAT(I3,2A2,I4,1X,I2,F4.2,A1,I3,F4.2,A1,I7,I3,I5,
44.  1 I1,I2,3I1,13X,I6)
45.  C  CHECKING APPARATUS CODE
46.  NAPP=IAPP+1
47.  GO TO (510,510,510,510,550,510) NAPP
48.  510 OUTPUT 'APPARATUS CODE NOT IMPLIMENTED'
49.  550 CONTINUE
50.  DELEV=FL9AT(IELEV)*.1
51.  IF(IELEVKEY.EQ.0.AND. IELEVC.NE.4) ELEV=DELEV; GO TO 560
52.  IDEP=DELEV

```

```

53.      DEP=FLBAT(IDEP)
54.      560 CONTINUE
55.      DLATM=DLATM/60.
56.      DLATD=FLBAT(ILATD)
57.      DLAT=DLATD+DLATM
58.      IF(NBRS.EG.IS) DLAT=-DLAT
59.      DLONGM=DLONGM/60
60.      DLONGD=FLBAT(ILONGD)
61.      DLONG=DLONGD+DLONGM
62.      IF(NBRS.EG.IWE) DLONG=-DLONG
63.      FA=FLBAT(IFA)*.1
64.      BG=FA*(.04185+DENS*DEP)
65.      ZG1=FLBAT(NG1)*1000.
66.      ZG2=FLBAT(NG2)*.01
67.      K977=NG1
68.      BBSG=ZG2
69.      KGHM=ISTA
70.      KGYR=IYR
71.      ENCODE(35,1005,IZ)IHSC,ISIGFA,ISIGBG,ILBC,ITR,ISER,IAPP,IELEV,
72.      1 IELEV
73.      1005 FORMAT(I4,2I2,I3,2A2,3I1)
74.      CALL UNPKBY(IZ,IW,35)
75.      DO 420 J=1,35
76.      IA(J)=ISL(IW(J),24)
77.      420 CONTINUE
78.      500 CONTINUE
79.      RLAT=DLAT*DEGRA;RLONG=DLONG*DEGRA
80.      CALL GINBT(ITAPE,UTAPE,K0,KGDA,KGMB,KGYR,KGHM,IDIF,ISBRC,
81.      1 RLAT,RLONG,ELEV,K977,BBSG,IDEF,FA,BG,
82.      2 TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
83.      NREC=NREC+1
84.      GO TO 50
85.      C
86.      C END OF FILE
87.      C
88.      995 CONTINUE
89.      ENDFILE UTAPE
90.      OUTPUT NREC
91.      OUTPUT 'ALL DONE'
92.      STOP
93.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
EG	R	SCALR	000A V	1	DEGRA	R	SCALR	000A6 V	1	DELEV	R	SCALR	000DA V	1
DENS	R	SCALR	000B V	1	DEP	R	SCALR	000B V	1	DLAT	R	SCALR	000DD V	1
DLATD	R	SCALR	000C V	1	DLATM	R	SCALR	000CB V	1	DLONG	R	SCALR	000DF V	1
DLONGD	R	SCALR	000D V	1	DLONGM	R	SCALR	000CE V	1	ELEV	R	SCALR	000AD V	1
FA	R	SCALR	000E V	1	FLBAT	R	SPR8G	000E V	1	GINGT	R	SPR8G	000AE V	1
I	I	SCALR	000F V	1	IA	I	ARRAY	000F V	35	INTRIN	I	SCALR	000B V	1
IDEP	I	SCALR	000G V	1	IDF	I	SCALR	000G V	1	IAPP	I	SCALR	000B3 V	1
IELEV	I	SCALR	000H V	1	IELEV	I	SCALR	000H V	1	IELEVKEY	I	SCALR	000B3 V	1
IELEVTV	I	SCALR	000I V	1	IELEV	I	SCALR	000I V	1	IFBC	I	SCALR	000C V	1
IFBC	I	SCALR	000J V	1	IFC	I	SCALR	000J V	1	IHSC	I	SCALR	000C2 V	1
IFIN	I	SCALR	000K V	1	IGC	I	SCALR	000K V	1	ILBC	I	SCALR	000C6 V	1
ILONGD	I	SCALR	000L V	1	ILATD	I	SCALR	000L V	1	IREGC	I	SCALR	000F V	1
IS	I	SCALR	000M V	1	ISL	I	SCALR	000M V	1	ISIGBG	I	SCALR	0004 V	1
ISIGFA	I	SCALR	000N V	1	ISL	I	SPR8G	000N V	1	ISRC	I	SCALR	000AE V	1
ISTA	I	SCALR	000O V	1	ISW	I	SPR8G	000O V	1	ITAPE	I	SCALR	000A1 V	1
ITR	I	SCALR	000P V	1	IW	I	ARRAY	000P V	35	IWE	I	SCALR	000A V	1
IYR	I	SCALR	000Q V	1	IZ	I	ARRAY	000Q V	9	J	I	SCALR	000A5 V	1
JDIF	I	SCALR	000R V	1	JELC	I	SCALR	000R V	1	JTAPE	I	SCALR	000A2 V	1
KGDA	I	SCALR	000S V	1	KGM	I	SCALR	000S V	1	KGM9	I	SCALR	000B V	1
KGYH	I	SCALR	000T V	1	KI	I	SCALR	000T V	1	KK	I	SCALR	000A8 V	1
KB	I	SCALR	000U V	1	KI977	I	SCALR	000U V	1	NAME	I	ARRAY	000A3 V	80
NAPP	I	SCALR	000V V	1	KREDA	I	SCALR	000V V	1	NGI	I	SCALR	000C V	1
NQ2	I	SCALR	000W V	1	KREDA	I	SCALR	000W V	1	NREC	I	SCALR	000AC V	1
NRSG	I	SCALR	000X V	1	KREDA	I	SCALR	000X V	1	RFA	R	SCALR	000BE V	1
RLAT	R	SCALR	000Y V	1	KREDA	I	SCALR	000Y V	1	STAT	R	SPR8G	000B V	1
TC	R	SCALR	000Z V	1	KREDA	I	SCALR	000Z V	1	ZG1	R	SCALR	000E V	1
ZG2	R	SCALR	0001 V	1	KREDA	I	SCALR	0001 V	1					

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
5C	0007B	550	000C0	510	000B3	560	000D2
99	0014C	1002	0012A	1003	00078	1005	00117

## LOCAL VARIABLES (228 WORDS):

0000	IA	00023	NAME	0007C	IM	0009F	IS	000A0	IWE
00001	ITAPE	000A2	JTAPE	000A4	IBUT	000A5	J	000A6	DEGRA
00007	RADEG	000A3	KI	000A4	K9	000AB	DENS	000AC	NREC
0000C	ELEV	000A5	KGDA	000A6	KGM9	000B1	KGYR	000B2	KGM
00009	JOIF	000A7	RLONG	000B6	K977	000B7	BSG	000B8	DEP
0000S	FA	000B7	TC	000B8	JELC	000BD	IGC	000BE	RFA
0000E	IREGC	000B9	IFBC	000C2	IMSC	000C3	ISIGFA	000C4	ISIGBG
0000I	I	000C0	IL9C	000C7	ISER	000C9	ISTA	000CA	ILATD
0000G	DLATM	000C1	ILONG	000C8	ILONGM	000CF	NEAPP	000D0	IELEV
00001	NGI	000C2	NGRS	000C9	IYR	000D5	IAPP	000D6	IELEV
00007	IELEVTV	000C3	IFBC	000D9	NAPP	000DB	DEP	000DC	DLATD

000DC DLAT (CODE DLONG 000DF DLONG 000E0 ZG1 000E1 ZG2 000E2 IDIF  
 000E3 IELC

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

FLBAT ISL

EXTERNAL SUBPROGRAMS REQUIRED:

GINOT	ISA	STAT	UNPKBY	F:101	F:103	F:105	F:108
M:DB	M:BC	9BCDRDEE	9BCDREAD	9ENCODE	9ENDFILE	9ENDIOL	9INITIAL
9I0DATA	9IT0R	9PRINT	9RT0I	9STOP			

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC	HEX
WORDS	WORDS
---	---
GENERATED CODE: 354	00162
CONSTANTS: 10	0000A
LOCAL VARIABLES: 228	000E4
TEMPS: 0	00000
---	---
TOTAL PROGRAM: 592	00250

(NO MEMORY PROTECTION)

```

1.  C   PROGRAM LSORT
2.  C   PROGRAM TO SORT AND EDIT LUNL O/P
3.  C   VERSION 29 APRIL 75 TO CHANGE TEST IN LINE 21
4.  C   VERSION OF 23 MAY 74 TO INCREASE RESOLUTION OF Z
5.  C   VERSION OF 29 MAR 74 TO MAKE BETTER EOF CHECK
6.  C   VERSION 1 MAR 74 TO STOP IF ISLCNT GT 20
7.  C   ORIGINAL VERSION 21 FEB 74
8.      ITAPE=0
9.      JTAPE=6
10.     KTAPE=13
11.     IIN=105
12.     OUTPUT 'PROGRAM LSORT VERSION OF 29 APRIL 75'
13.     READ(IIN,1004) ZLIM
14.     OUTPUT ZLIM
15.     READ(IIN,1001) NUMBED
16.     DO 500 I=1,NUMBED
17.         ITAPE=ITAPE+1
18.         JTAPE=JTAPE+1
19.         ISLCNT=0
20.         ZST=999.0
21. 7C   READ(ITAPE,1002,END=400) NCNT,RH0,Z
22.         IF(Z.EQ.0.0.AND.ISLCNT.NE.0.AND.RH0.EQ.0.0) GO TO 400
23.         ZCHK=ABS(Z-ZST)
24.         IF(ZCHK.LT.ZLIM) GO TO 600
25.         ZST=Z
26.         ISLCNT=ISLCNT+1
27.         IF(ISLCNT.GT.20) OUTPUT 'TOO MANY LANIMALS, ISLCNT,I, GO TO 999
28.         WRITE(JTAPE,1002) NCNT,RH0,Z
29. 8C   READ(ITAPE,1003) X,Y,LSLPT
30.         WRITE(JTAPE,1003) X,Y,LSLPT
31.         IF(LSLPT.NE.1) GO TO 80
32.         GO TO 7C
33. 40C  WRITE(KTAPE,1001) ISLCNT
34. 50C  CONTINUE
35.         GO TO 999
36. 60C  CONTINUE
37.         READ(ITAPE,1003) X,Y,LSLPT
38.         IF(LSLPT.EQ.1) GO TO 7C
39.         GO TO 60C
40. 999  STOP
41.  C   *****
42.
43.  C   FORMATS
44.  C   *****
45. 1001 FORMAT(I2)
46. 1002 FORMAT(I2,F10.4,F16.6,F6.3)
47. 1003 FORMAT(2F12.5,I1)
48. 1004 FORMAT(F10.0)
49.     END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
ABS	R	SPRNG	-----	-----	I	-----	-----	-----	-----	IIN	I	SCALR	00003	V
ISLCNT	I	SCALR	00007	V	ITAPE	I	SCALR	00000	V	JTAPE	I	SCALR	00001	V
KTAPE	I	SCALR	00002	V	LSLPT	I	SCALR	0000F	V	NCNT	I	SCALR	00009	V
NUMB9D	I	SCALR	00005	V	RHB	R	SCALR	0000A	V	X	R	SCALR	0000D	V
Y	R	SCALR	0000F	V	Z	R	SCALR	0000B	V	ZCHK	R	SCALR	0000C	V
ZLIM	R	SCALR	00004	V	ZST	R	SCALR	00008	V					

HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL
70	00034	80	00069	400	0007B	500	00080
1001	00093	1002	0009F	1003	0009C	1004	000A0

## LOCAL VARIABLES (16 WORDS):

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
00000	ITAPE		00001	JTAPE	00002	KTAPE		00003	IIN
00006	I		00007	ISLCNT	00008	ZST		00009	NCNT
0000C	ZCHK		0000D	X	0000E	Y		0000F	LSLPT

## BLANK COMMON (0 WORDS)

## INTRINSIC SUBPROGRAMS USED:

ABS

## EXTERNAL SUBPROGRAMS REQUIRED:

F:101	F:102	F:103	F:104	F:105	F:106	F:108	M:09
M:0C	9BCDDEE	9BCDDEE	9BCDARIT	9ENDI9L	9INITIAL	9I8DATA	9PRINT
9ST9P							

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
165	000A5
1	00001
16	00010
0	00000
182	000B6

GENERATED CODE:

CONSTANTS:

LOCAL VARIABLES:

TEMPS:

TOTAL PROGRAM:

```

1. C PROGRAM MODPLOT
2. C PLOTS DATA FOR PREPARATION OF, AND CONSTRUCTED,
3. C STRUCTURE MODELS OF EARTH'S CRUST
4. C MAKES PLOTS OF OUTPUT FROM TALPLOT 15, PROJ4, AND/OR SAINT2
5. CHRIS WOODING VERSION OF 4 NOV 1975
6. C VERSION 7 APRIL 75 TO ZERO VARIABLES
7. C VERSION 24 FEB 1975, TO ADD USE OF PINOT AND YINOT
8. C VERSION OF 3 FEB 1975, SO GINOT WILL NOT TRY TO READ EOTF SN CARDS
9. C VERSION OF 11 DEC 1974, TO ADD HANDLING OF PROJ4 OUTPUT TO 9T TAPE
10. C VERSION OF 26 SEPT 1974 TO CHANGE AND IMPROVE DOCUMENTATION
11. C VERSION OF 4 FEB 1973, TO CHANGE GSUM READ TO DECIMAL DEGREES
12. C VERSION OF 12 OCT 1972, TO CORRECT ERROR IN PLOTTING
13. C POLYGON MODEL FROM CARDS WITH JFMT = 5
14. C VERSION OF 8 SEPT 1972, TO UPDATE SEISMICITY INPUT FORMAT
15. C VERSION OF 7 SEPT 1972, TO CHANGE FORMAT OF PCS CARD
16. C VERSION OF 6 APRIL 1971, TO PLOT 2-D BOUGUER ANOMALY FROM TALPLOT
17. C DIMENSION IBUF(1000), LABEL(20)
18. C DIMENSION CXL(100),CYL(100),BGA(200),FX(200),SSELZ(200)
19. C DIMENSION DWGT(200)
20. C DIMENSION KSW(80),FZ(200)
21. C DIMENSION IDESC(6),VEL(8),THICK(8)
22. C DIMENSION BG2D(200)
23. C DIMENSION JA(10),JB(30)
24. C DIMENSION IA(35)
25. C DATA NNS,NEW,IS 1,1W 1/
26. C
27. C WHEN USING OUTPUT FROM PROJ4 ON MAG TAPE,
28. C JFMT NUMBER CARDS MUST STILL BE INPUT ON CARDS.
29. C IF JFMT=1 AND DATA IS ON MAG TAPE, THEN EITP CARDS HAVE TO BE
30. C INCLUDED FOR SUBROUTINE MOUNT, FOLLOWING THE JFMT=1 CARD
31. C
32. C
33. C SSW(1)=0 TO PLOT ONLY FROM TALPLOT OUTPUT TAPE
34. C #1 TO PLOT PROJ4 DATA, INCLUDING MODEL POLYGONS
35. C #2 TO PLOT BOTH TALPLOT OUTPUT TAPE AND PROJ4 DATA
36. C SSW(2)=1 TO PLOT BOUGUER ANOMALY IN ADDITION TO THE FREE-AIR
37. C SSW(3)=1 TO PLOT HEIGHT FROM GSUM DATA
38. C SSW(4)=1 TO PLOT ELEVATION, (INPUT VALUES ARE IN METERS)
39. C SSW(5)=1 TO PLOT OBSERVED + CALC. GRAVITY
40. C SSW(6)=1 TO PLOT WEIGHT
41. C SSW(7)=1 TO PLOT CONTR. BUTION OF EACH POLYGON
42. C SSW(8)=1 TO PLOT 2-D BOUGUER ANOMALY IN TALPLOT OUTPUT
43. C SSW(13)=1 TO PRINT INTERMEDIATE VALUES
44. C SSW(14)=1 TO PLOT ONLY A DOT FOR GSUM FREE-AIR VALUES, RATHER
45. C THAN A CONTINUOUS LINE
46. C SSW(30)=1 TO READ GSUM DATA ON 2 CARDS
47. C SSW(32)=1 TO READ SPFMT DATA ON 2 CARDS
48. C SSW(36)=1 TO READ SEISMICITY DATA FROM CARDS
49. C
50. C
51. C USES INCEP, EXT0, ISW ,SPL0T, SP0T,GIN0T,DISAZ
52. C
53. C INITIALIZATION SECTION
54. C
55. C CALL STAT
56. C RFACT=1
57. C I=ISW(=2)
58. C SETTING SSW(31)=1 SO GINOT WILL NOT TRY TO READ EOTF SERIAL NO. CARDS
59. C II=ICHG(31,1)

```

```

60.      CALL PLOTS(IBUF,=1000)
61.      OUTPUT 'MODPLOT VERSION OF 4 NOV 1975'
62.      1 IIN=105
63.      IOUT=108
64.      ITAPE=1
65.      JTAPE=2
66.      IGSUM=0
67.      L=99
68.      KFXN=1
69.      IL00P=0
70.      DEGRA=1.745329E+02
71.      RADEG=57.29578
72.      I00B = 0
73.      JFMT=0
74.      IYIN = 0
75.      IPIN = 0
76.      DO 100 I=1,200
77.      0GA(I)=0.0
78.      FX(I)=0.0
79.      FZ(I)=0.0
80.      SSELZ(I)=0.0
81.      DWGT(I)=0.0
82.      BG2D(I)=0.0
83.      100 CONTINUE
84.      AI=0.0
85.      AS=0
86.      ANGB=0.0
87.      DO 101 I=1,100
88.      CXL(I)=0.0
89.      CYL(I)=0.0
90.      101 CONTINUE
91.      DISTKM=0.0
92.      DO 102 I=1,8
93.      VEL(I)=0.0
94.      THICK(I)=0.0
95.      102 CONTINUE
96.      J1=0
97.      J2=0
98.      J3=0
99.      J4=0
100.     J5=0
101.     J6=0
102.     J7=0
103.     J8=0
104.     K1=0
105.     K2=0
106.     K3=0
107.     K4=0
108.     K5=0
109.     K6=0
110.     K7=0
111.     K8=0
112.     C
113.     C XFACT = NUMBER KM'S/INCH IN X DIRECTION (LONG AXIS OF PLOT)
114.     C YFACT = NUMBER OF KM'S IN Y DIRECTION
115.     C XWIDE = KM WIDTH OF PLOT IN X DIRECTION
116.     C YWIDE = KM DEPTH OF PLOT IN Y DIRECTION
117.     C TOP,BOT,BLEFT,RIGHT, = KM VALUE FOR THOSE
118.     C

```

```

120. C          ABOVE SEA LEVEL = NEGATIVE
121. C          FOR OTHER CURVES THAN MODEL,  EDPH = NEGATIVE,  ELEV = POSITIVE
122. C
123. C          BOUNDARIES OF MGD PLOT
124. C          ELFAC,GFAC,WFAC,PFAC,---ARE THE SCALE FACTORS FOR ELEVATION
125. C          (KMS/IN),GRAVITY(MGAL/IN),WEIGHT(KG/IN) AND
126. C          INDIVIDUAL POLYGON CONTRIBUTION (MG/IN).1
127. C          ELDIS,GDIS,PDIS,---ARE THE DISTANCE IN INCHES OF THE
128. C          ORIGINS OF THE CURVES ABOVE THE ORIGINS OF THE MODEL
129. C          WDIS,---DISTANCE OF WEIGHT CURVE BELOW BOTTOM OF MODEL
130. C          ORIGIN OF MODEL IN Y DIRECTION = DBOT + WDIS + (BOT/YFAC)
131. C
132. C          READ(IIN,11)XFACT,YFACT,TOP,BOT,BLEFT,RIGHT
133. 11  FORMAT(8F10.2)
134. C          OUTPUT XFACT,YFACT,TOP,BOT,BLEFT,RIGHT
135. C          TOP=TOP
136. C          BOT=BOT
137. C          ICHAR=0
138. C          ISTR=0
139. C          READ(IIN,11) ELFAC,ELDIS,GFAC,GDIS,WFAC,WDIS,PFAC,PDIS
140. C          READ(IIN,11) HT,DBOT
141. C HT= CHARACTER HEIGHT MULTIPLICATION FACTOR (USED IN THE CALL TO
142. C          SYMBOL FOR THE PLOTTING OF THE ANOMALY CURVES
143. C          IF HT IS EQUAL TO ZERO A DEFAULT VALUE OF 3 IS ASSUMED
144. C          DBOT IS THE DISTANCE THAT THE WEIGHT CURVE IS SUPPOSED TO BE
145. C          PLOTTED ABOVE THE BOTTOM OF THE PLOT (RIGHT SIDE OF PLOTTER)
146. C          OUTPUT ELFAC,ELDIS,GFAC,GDIS,WFAC,WDIS,PFAC,PDIS,HT
147. C          1 ,DBOT
148. C          IF(HT.EQ.0) HT=3.
149. C          HT=HT*0.035
150. C          XWIDE = RIGHT-BLEFT
151. C          YWIDE = TOP-BOT
152. C          SL = (XWIDE/XFACT)*0.5
153. C          SW = (YWIDE/YFACT)*0.5
154. C          CALL WHERE(XORG,YORG,RFACT)
155. C          CALL PLOT(XORG,YORG,*3)
156. C          CALL SYMBOL(0.0,0.0,0.28,9,0.0,2)
157. C          IF(ISH(1)=1,5019,5018,5019)
158. 5018 INPT=105
159. C          GO TO 5021
160. 5019 INPT=1
161. 5021 READ(INPT,5022) LABEL
162. 5022 FORMAT(20A4)
163. C          WRITE(IIOU,5023) LABEL
164. 5023 FORMAT(1X,20A4)
165. C          CALL SYMBOL(0.0,1.0,0.28,LABEL,90.0,80)
166. C VIT=DISTANCE OF MODEL ORIGIN ABOVE BOTTOM (RIGHT) OF PAPER
167. C          VIT=DBOT+WDIS+YWIDE/YFACT
168. C          IF(VIT.GE.29.) OUTPUT (PLOT TOO WIDE, WIDTH=VIT,OUTPUT VIT)
169. C          STOP
170. C          IF(BLEFT)5025,5024,5024
171. 5024 XT=3.0
172. C          GO TO 5026
173. 5025 XT=3.0*(BLEFT/XFACT)
174. 5026 CALL PLOT(XT,VIT,*3)
175. C          XX = 0.0
176. C          YY = 0.0
177. C          CALL SYMBOL(0.0,0.0,0.14,9,0.0,1)
178. C CALCULATING COORDINATES OF CENTRAL POINT
179. C          RX = (XWIDE*0.5) - (1.0*BLEFT)

```

```

180.      RY = -1.0*((YWIDE*0.5)-(TOP))
181.  C   CONVERTING ORIGIN TRANSLATION VECTORS TO INCHES
182.      RX = RX/XFACT
183.      RY = RY/YFACT
184.  C   OUTPUT INTERMEDIATE VALUES
185.      WDIS = -YWIDE/YFACT-WDIS
186.      IF(ISW(13)) 402,698,402
187. 402  WRITE(IIBUT,404)XWIDE,YWIDE,SL,SW,RX,RY,XFACT,YFACT
188. 404  FORMAT(11V= 1,2E12.5/2E12.5/2E12.5/2E12.5)
189.  C   PLOT ORIGIN IS AT 0,0 OF STRUCTURE MODEL
190. 698  CONTINUE
191.  C   INITIALIZATION IS COMPLETE
192.  C
193.      IF(ISW(1))115,115,405
194.  C   READ CSS VALUES (FIRST DATA CARD USED IN PROJ4)
195. 405  READ(IIN,699)JA,ANG,DMAXM,ILAT,RILTM,ILONG,RILOM,JB
196. 699  FORMAT(10A1,2F10.0,14,F6.2,14,F6.2,30A1)
197.      WRITE(IIBUT,699)JA,ANG,DMAXM,ILAT,RILTM,ILONG,RILOM,JB
198.      RILT=DMT0R(ILAT,RILTM)
199.      RILG=DMT0R(ILONG,RILOM)
200.      IF(ISW(1)=2)700,115,700
201.  C
202.  C   READING TALPLOT FORMAT
203.  C
204.  C
205. 115  CONTINUE
206.      READ(ITAPE,501)KSW
207. 501  FORMAT(80I1)
208.      READ(ITAPE,120) RDENS,RWGT,RHOD,REFX,FXI,DELFx,M,IMAX
209. 120  FORMAT(6F10.2,2I10)
210.      READ(ITAPE,135)(OGA(I),I=1,M)
211. 135  FORMAT(5F10.1)
212.      IF(KSW(2))553,553,552
213. 552  CONTINUE
214.      READ(ITAPE,135) (FZ(I),I=1,M)
215. 553  CONTINUE
216.  C
217.  C   NOW START READING POLYGONS
218.  C
219. 145  READ(ITAPE,150) LNO,RHORK
220. 150  FORMAT(15,F10.3)
221. 160  I = 1
222. 162  CONTINUE
223.      READ(ITAPE,168)XX,YY,ICODE
224. 168  FORMAT(2F10.2,I1)
225.  C   MAKING DEPTHS NEGATIVE
226.      YY = -YY
227.  C   CONVERTING COORDINATES TO INCHES
228.      XX = XX/XFACT
229.      YY = YY/YFACT
230.  C   SHIFTING COORDINATES TO CENTER OF PLOT
231.      CXL(I) = XX-RX
232.      CYL(I) = -1.0*(RY-YY)
233.      CX = CXL(I)
234.      CY = CYL(I)
235.      IF(I-1)190,190,200
236. 190  CONTINUE
237.      GO TO 250
238. 200  CXP = CXL(I-1)

```

```

240.      CALL INCEP(SL,SW,CX,CY,CXP,CYP,BX,BY,A1,A2,B1,B2)
241.      IF(ISW(13))406,205,406
242.      406 WRITE(IIOUT,408)SL,SW,CX,CY,CXP,CYP,BX,BY,A1,A1,B1,B2
243.      408 FORMAT(IIV=1,6E10.5/6E10.5)
244.      205 IF(A1-9990.0)210,248,248
245.      C PLOT MODEL SEGMENT
246.      210 XX = A1+RX
247.      YY = B1+RY
248.      IPEN = 3
249.      CALL PLOT(XX,YY,IPEN)
250.      XX = A2+RX
251.      YY = B2+RY
252.      IPEN = 2
253.      CALL PLOT(XX,YY,IPEN)
254.      248 IF(ICODE=9)250,260,250
255.      250 I=I+1
256.      GO TO 162
257.      260 IF((JFMT.EQ.5) .OR. (JFMT.EQ.6)) GO TO 272
258.      262 NCDE = 1
259.      C CHECKING IF POLYGON NUMBER = 1
260.      IF(LN=1)1260,1190,1260
261.      C CHECK IF PLOT 2-D BOUGUER ANOMALY
262.      1190 IF(ISW(8))1260,1260,1200
263.      1200 DO 1220 II=KFXN,M
264.      READ(ITAPE,1206)K,FX(K),SSELZ(K),BG2D(K)
265.      1206 FORMAT(15,F10.2,10X,2F10.2)
266.      C THIS SECTION PLOTS INTERMEDIATE DATA FOR EACH POLYGON
267.      IF(ISW(7))1210,1212,1210
268.      1210 XX=FX(K)/XFACT
269.      YY=SSELZ(K)/PFAC+PDIS
270.      YPB=YY+VIT
271.      IF(YPB.GE.29 .OR. YPB.LE.0) NCDE=1; GO TO 1220
272.      CALL SYMBOL(XX,YY,HT,ICHAR,0.0,NCDE)
273.      1212 CONTINUE
274.      NCDE = 2
275.      1220 CONTINUE
276.      GO TO 271
277.      1260 IF(ISW(1).EQ.1) GO TO 271
278.      IF((JFMT.EQ.5) .OR. (JFMT.EQ.6)) GO TO 271
279.      C
280.      C      NEXT TWO IF STATEMENTS ASSUME PLOTTING OF MODEL POLYGONS
281.      C      IS ONLY BEING DONE FROM TALPLOT OUTPUT ON MAG TAPE
282.      C
283.      C ILOOP IS .EQ. 1 ONLY WHEN WE ARE PLOTTING THE MODIFIED POLYGON
284.      IF(ILOOP.EQ.1) GO TO 302
285.      IF(KSW(9).EQ.1) GO TO 271
286.      DO 270 III=KFXN,M
287.      READ(ITAPE,263)K,FX(K),SSELZ(K)
288.      263 FORMAT(15,F10.2,10X,F10.2)
289.      C THIS SECTION PLOTS INTERMEDIATE DATA FOR EACH
290.      C POLYGON
291.      IF(ISW(7))554,555,554
292.      554 XX=FX(K)/XFACT
293.      YY=SSELZ(K)/PFAC+PDIS
294.      YPB=YY+VIT
295.      IF(YPB.GE.29 .OR. YPB.LE.0) NCDE=1; GO TO 270
296.      CALL SYMBOL(XX,YY,HT,ICHAR,0.0,NCDE)
297.      555 CONTINUE
298.      NCDE = 2
299.      270 CONTINUE

```

```

300. 271 CONTINUE
301. ICHAR=ICAR+1
302. 272 IF(LN8=L) 145,290,145
303. 290 CONTINUE
304. IF(ISW(1))291,291,700
305. 291 IF(KSW(6).EQ.0) GO TO 300
306. IL88P=1
307. GO TO 160
308. 300 IF((JFMT.EQ.5) .OR. (JFMT.EQ.6)) GO TO 330
309. 302 DO 310 K=KFXN,M
310. READ(ITAPE,304)J,FX(K),SSELZ(K),DWGT(K)
311. 304 FORMAT(I5,F10.2,20X,F10.2,36X,F16.0)
312. 310 CONTINUE
313. C NOW PLOT ELEV,OGA,WEIGHT,SSELZ,AS REQUIRED BY SENSE
314. C SWITCHES
315. NCDE = -1
316. IF(ISW(4))556,557,556
317. 556 DO 590 K=KFXN,M
318. XX=FX(K)/XFACT
319. YY=FZ(K)*0.001/ELFAC+ELDIS
320. YP8=YY+VIT
321. IF(YP8.GE.29 .OR. YP8.LE.0) NCDE=-1; GO TO 590
322. CALL SYMBOL(XX,YY,HT,0,0,0,NCDE)
323. NCDE=-2
324. 590 CONTINUE
325. 557 CONTINUE
326. C NOW PLOT OBS. ERFEP + CALCULATED G
327. IF(ISW(5))558,559,558
328. 558 NCDE = -1
329. DO 599 K=KFXN,M
330. XX=FX(K)/XFACT
331. YY=SSELZ(K)/GFAC+GDIS
332. YP8=YY+VIT
333. IF(YP8.GE.29 .OR. YP8.LE.0) NCDE=-1; GO TO 599
334. 591 CALL SYMBOL(XX,YY,HT,11,0,0,NCDE)
335. NCDE=-2
336. 599 CONTINUE
337. NCDE = -1
338. IPEN=3
339. DO 592 K=KFXN,M
340. XX=FX(K)/XFACT
341. YY=OGA(K)/GFAC+GDIS
342. YP8=YY+VIT
343. IF(YP8.GE.29 .OR. YP8.LE.0) NCDE=-1; IPEN=3; GO TO 592
344. C CALL SYMBOL(XX,YY,HT,0,0,0,NCDE)
345. CALL PLOT(XX,YY,IPEN)
346. NCDE=-2
347. IPEN=2
348. 592 CONTINUE
349. 559 CONTINUE
350. C
351. C CHECK IF PLOT 2-D BOUGUER ANOMALY
352. C IF SO, PLOT LINE WITHOUT SYMBOL
353. C
354. IF(ISW(8)) 588,588,570
355. 570 IPEN=3
356. DO 585 K=KFXN,M
357. XX=FX(K)/XFACT
358. YY=BG2D(K)/GFAC + GDIS

```

```

360.      IF(YP0.GE.29. OR. YP0.LE.0) IPEN=3; GO TO 585
361.      CALL PLOT(XX,YY,IPEN)
362.      IPEN=2
363.      585 CONTINUE
364.      588 NCDE = -1
365.      IF(ISW(6))562,562,561
366.      561 DO 593 K=KFXN,M
367.      XX=FX(K)/XFACT
368.      YY=DWGT(K)/WFAC+WDIS
369.      YP0=YY+VIT
370.      IF(YP0.GE.29 .OR. YP0.LE.0) NCDE=-1; GO TO 593
371.      CALL SYMBOL(XX,YY,HT,1.0,0,NCDE)
372.      NCDE=-2
373.      593 CONTINUE
374.      562 CONTINUE
375.      330 WRITE(IIOUT,332)
376.      332 FORMAT('LAST POLYGON OF MODEL PLOTTED')
377.      IF(ISW(1))333,333,700
378.      333 CALL PLOT(XX,YY,999)
379.      CALL EXIT
380.      700 ITAPE = 105
381.      C  SETTING UP CONSTANTS FOR PLOTTING SPFMT COLUMNS
382.      C  BDIST = LENGTH OF TICK LINE IN INCHES
383.      C  ANGB = ANGLE IN DEGREES OF TICK LINE FROM HORIZONTAL
384.      C          UPWARD ANGLE = + ANGLE
385.      C          DOWNWARD ANGLE = - ANGLE
386.      C  YFAC = KM PER INCH FOR PLOTTING COLUMN
387.      C  SEP = SEPARATION DISTANCE IN INCHES BETWEEN COLUMNS
388.      DEGRA = 1.745329E-2
389.      RADEG = 57.29578
390.      RANGB = ANGB * DEGRA
391.      AC = COS(RANGB)
392.      AS = SIN(RANGB)
393.      AS=-1.0*AS
394.      KDA = 0
395.      KMB = 0
396.      KYR = 0
397.      BDIST=0.25
398.      ANGB=0.0
399.      YFAC=YFACT
400.      KGDA=0
401.      KGMB=0
402.      KGYR=0
403.      KGHM=0
404.      ZHT=1.0
405.      HGT=0.07
406.      C
407.      C *****
408.      C  JFMT = 1 FOR GSUM FORMAT
409.      C  JFMT = 2 FOR GSUM FORMAT
410.      C  JFMT = 3 FOR SPFMT DATA
411.      C  JFMT = 4 FOR SEISMICITY DATA
412.      C  JFMT = 5 FOR MODEL POLYGONS
413.      C  JFMT = 6 FOR TALPLOT INPUT
414.      C  JFMT = 9 TO TERMINATE JOB
415.      C *****
416.      701 READ (IIN,702)JFMT
417.      702 FORMAT(I1)
418.      703 OUTPUT JFMT
419.      GO TO (710,710,750,800,145,115,333,333,996)JFMT

```

```

420. C PLOTTING GSUM DATA
421. 710 CONTINUE
422.     IPEN=3
423. 712 CONTINUE
424.     IF(ISW(30).EQ.0)ITAPE=1,G0 TO 713
425. C
426. C INITIALIZE GINOT IF THIS IS THE FIRST READ FOR GINOT
427. C
428. 713 IF(IGSUM.EQ.1)G0 TO 714
429.     KK=0
430.     CALL GINOT(ITAPE,UTAPE,KK,KGDA,KGM0,
431. 1 KGYR,KGHM,IDIF,ISORC,RLAT,RLONG,ELEV,K977,0BSG,
432. 2 IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
433.     IGSUM=1
434. 714 KK=1
435.     CALL GINOT(ITAPE,UTAPE,KK,KGDA,KGM0,
436. 1 KGYR,KGHM,IDIF,ISORC,RLAT,RLONG,ELEV,K977,0BSG,
437. 2 IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
438.     IF(KK.EQ.8)G0 TO 700
439.     IF(KK.EQ.9)G0 TO 999
440. 63 KGDA0=KGDA
441.     KGM00=KGM0
442.     KGYR0=KGYR
443.     KGHM0=KGHM
444. 73 DEPTH=IDEP
445.     IF(IDEP)78,74,78
446. 74 HEIGHT=ELEV
447.     G0 TO 64
448. 78 HEIGHT=-DEPTH
449. 64 CONTINUE
450. C CONVERTING HEIGHT FROM METERS TO KM
451.     HEIGHT=HEIGHT*0.001
452. C DETERMINE DISTANCE FROM ORIGIN
453.     G0 TO 40
454. 720 YY=GDIS+(FA/GFAC)
455.     XX=DISTKM/XFACT
456.     YP0=YY+VIT
457.     IF(YP0.GE.29. OR. YP0.LE.0) IPEN=3,G0 TO 723
458.     IF(ISW(14))721,721,722
459. C PLOTTING A SMALL CIRCLE FOR FREE-AIR
460. 721 CALL PLOT(XX,YY,IPEN)
461.     IPEN=2
462.     G0 TO 723
463. C PLOTTING ONLY A DOT FOR FREE-AIR
464. 722 CALL PLOT(XX,YY,3)
465.     CALL PLOT(XX,YY,2)
466.     CALL PLOT(XX,YY,3)
467.     G0 TO 723
468. C CHECKING IF ALSO PLOT BOUGUER ANOMALY
469. 723 IF(ISW(2))728,728,725
470. C PLOT BOUGUER ANOMALY
471. 725 XT=XX
472.     YT=GDIS+(BG/GFAC)
473.     YP0=YT+VIT
474.     IF(YP0.GE.29. OR. YP0.LE.0) G0 TO 728
475.     CALL PLOT(XT,YT,3)
476.     CALL SPOT(XT,YT)
477.     CALL PLOT(XX,YY,3)
478. C CHECKING IF ALSO PLOT HEIGHT

```

```

480. 730 XT=XX
481.    YT=HEIGHT/ELFAC
482.    YP0=YT+VIT
483.    IF(YP0,GE.29. OR. YP0,LE.0) GO TO 712
484.    CALL PLOT(XT,YT,3)
485.    CALL SPOT(XT,YT)
486.    CALL PLOT(XX,YY,3)
487.    GO TO 712
488. C PLOTTING SPFMT DATA
489. 750 IF(ISH(32).EQ.0) ITAPE =1
490. CC
491. C READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
492. C
493. C INITIALIZING PINOT, IF THIS IS FIRST READ
494.   IF (IPIN.EQ.1) GO TO 752
495.   KK = 0
496.   CALL      PINOT(ITAPE, JTAPE, KK, ISTA, KEY, LAT, LATM, KNS,
497. 1  LONG, LOM, KEW, VEL, THICK, IMANT, NELEV, N1, N2, N3, N4, MET, IYR, IDESC,
498. 2  DINE, STHIK, CRVN, WGTN, AVWTN, CRVW, WGTW, AVWTW)
499.   IPIN = 1
500. 752 KK=1
501.   CALL      PINOT(ITAPE, JTAPE, KK, ISTA, KEY, LAT, LATM, KNS,
502. 1  LONG, LOM, KEW, VEL, THICK, IMANT, NELEV, N1, N2, N3, N4, MET, IYR, IDESC,
503. 2  DINE, STHIK, CRVN, WGTN, AVWTN, CRVW, WGTW, AVWTW)
504. 18 IF(KK.EQ.8) GO TO 700
505.   IF(KK.EQ.9) GO TO 999
506. 20 CONTINUE
507.   VEL(1)=(FLOAT(J1))*0.1
508.   VEL(2)=(FLOAT(J2))*0.1
509.   VEL(3)=(FLOAT(J3))*0.1
510.   VEL(4)=(FLOAT(J4))*0.1
511.   VEL(5)=(FLOAT(J5))*0.1
512.   VEL(6)=(FLOAT(J6))*0.1
513.   VEL(7)=(FLOAT(J7))*0.1
514.   VEL(8)=(FLOAT(J8))*0.1
515.   THICK(1)=(FLOAT(K1))*0.1
516.   THICK(2)=(FLOAT(K2))*0.1
517.   THICK(3)=(FLOAT(K3))*0.1
518.   THICK(4)=(FLOAT(K4))*0.1
519.   THICK(5)=(FLOAT(K5))*0.1
520.   THICK(6)=(FLOAT(K6))*0.1
521.   THICK(7)=(FLOAT(K7))*0.1
522.   THICK(8)=(FLOAT(K8))*0.1
523.   VMANT=(FLOAT(IMANT))*0.1
524.   ELEV=NELEV
525.   ELEV=ELEV*0.01
526. 50 IF(N1.2) 70,60,70
527. C SEA SEISMIC PROFILE
528. 60 VELW= 1.5
529.   WATTK= ELEV
530.   GO TO 80
531. C LAND SEISMIC PROFILE
532. 70 VELW= 0.0
533.   WATTK= 0.0
534. C MAIN PLOTTING Loop
535. 80 RLATM = LATM
536.   RLOM = LOM
537.   RLAT = SDMTOR(LAT,RLATM)
538.   RLONG = SDMTOR(LONG,RLOM)
539.   IF(KNS=NNS)54, 52, 54

```

```

540.      52  RLAT = RLAT
541.      54  IF (KEW=NEW) 58,56,58
542.      56  RLONG = RLONG
543.      58  CONTINUE
544.      C  DETERMINE DISTANCE FROM ORIGIN
545.          GO TO 40
546.      451  XX=DISTKM/XFACT
547.          YY = 0.0
548.          CALL PLOT (XX,YY,3)
549.          CALL SPLOT (ISTA,RLAT,RLONG,VEL,THICK,VELW,WATTK,VMANT,XX,YY)
550.          1  YFAC,ZHT,HGT,AC,AS,ANGB,BDIST)
551.          GO TO 750
552.      C  PLOTTING SEISMICITY DATA
553.      800  CONTINUE
554.          IF (ISW(36).EQ.0) ITAPE=1
555.      C
556.      C  INITIALIZING YINOT, IF THIS IS FIRST READ
557.          IF (IYIN.EQ.1) GO TO 801
558.          KK = 0
559.          CALL YINOT (ITAPE,JTAPE,KK,
560.          1  ISR1,ISR2,KDA,KM0,KYR,KHM,SEC,DLAT,KSND,DLON,KWE,DEPT,AMAG,IMB,
561.          2  IS0S,INTS,IDIAS,ITSU,ISEICH,IVOLC,INONT,ING,IFEG,IMS,IASP,IZH,
562.          3  ICE,IMG,IAUTH,IQHY,NPP,I0Q,ILM,IS1,IS2)
563.      C
564.          IYIN = 1
565.      801  KK = 1
566.          CALL YINOT (ITAPE,JTAPE,KK,
567.          1  ISR1,ISR2,KDA,KM0,KYR,KHM,SEC,DLAT,KSND,DLON,KWE,DEPT,AMAG,IMB,
568.          2  IS0S,INTS,IDIAS,ITSU,ISEICH,IVOLC,INONT,ING,IFEG,IMS,IASP,IZH,
569.          3  ICE,IMG,IAUTH,IQHY,NPP,I0Q,ILM,IS1,IS2)
570.          IF (KK.EQ.8) GO TO 700
571.          IF (KK.EQ.9) GO TO 999
572.      C  DETERMINE DISTANCE FROM ORIGIN
573.          KGDA=KDA
574.          KGMB=KM0
575.          KGYR=KYR
576.          KGHM=KHM
577.          KGDA0=KGDA
578.          KGMB0=KGMB
579.          KGYR0=KGYR
580.          KGHM0=KGHM
581.          KL=0
582.          CALL DNAV (DLAT,KSND,DLON,KWE,RLAT,RLONG,KL)
583.          GO TO 40
584.      820  XX=DISTKM/XFACT
585.          YY=DEPT/YFACT
586.          YPB=YY*VIT
587.          IF (YPB.GE.29. OR. YPB.LE.0) GO TO 800
588.          CALL PLOT (XX,YY,3)
589.          CALL ANGV3 (XX,YY,DEPT,AMAG)
590.          GO TO 800
591.      40  CONTINUE
592.          CALL DISAZ (RLAT,RLONG,RILT,RILG,1,A,B,DISTKM,C)
593.          IF (ANG) 44,44,42
594.          42  IF (A=135) 48,48,46
595.          44  IF (A=225) 48,48,46
596.          46  DISTKM=DISTKM
597.          48  CONTINUE
598.          IF (DISTKM.LT.BLEFT. OR. DISTKM.GT.RIGHT) I00B = I00B + 1
599.          GO TO (720,720,451,820) JFMT

```

```
600.      49  OUTPUT '00B'  
601.      GO TO (712,712,752,800)JFMT  
602.      996  WRITE(IIOUT,997)  
603.      997  FORMAT('JFMT = 91')  
604.      998  FORMAT(1H0,'END OF PROCESSING',/,  
605.      *      15,'DATA POINTS OUT OF PLOT BOUNDS')  
606.      999  WRITE(IIOUT,998) I00B  
607.      GO TO 333  
608.      END
```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
A	R	SCALR	00A2A V	1	AC	R	SCALR	00A8F V	1	AI	R	SCALR	00A38 V	1
AMAG	R	SCALR	00ADA V	1	ANG	R	SCALR	00A6B V	1	ANGB	R	SCALR	00A3A V	1
ANOV3	R	SPRGG	EXTERN	1	AS	R	SCALR	00A39 V	1	AVHTN	R	SCALR	00AC8 V	1
AVNTH	R	SCALR	00ACB V	1	AT	R	SCALR	00A84 V	1	A2	R	SCALR	00A85 V	1
B	R	SCALR	00AF3 V	1	B0IST	R	SCALR	00A93 V	1	BG	R	SCALR	00AA5 V	1
B2CD	R	ARRAY	00312 V	200	BLEFT	R	SCALR	00A50 V	1	B0T	R	SCALR	00A4F V	1
BX	R	SCALR	00A82 V	1	BY	R	SCALR	00A83 V	1	B1	R	SCALR	00A86 V	1
B2	R	SCALR	00A87 V	1	C	R	SCALR	00AF4 V	1	C0S	R	SPRGG	INTRIN	1
CRVN	R	SCALR	00AC6 V	1	CRVM	R	SCALR	00AC9 V	1	CX	R	SCALR	00A7E V	1
CYL	R	ARRAY	003FC V	100	CYP	R	SCALR	00A80 V	1	CY	R	SCALR	00A7F V	1
DEGRA	R	ARRAY	00460 V	100	DELFX	R	SCALR	00A81 V	1	DEBT	R	SCALR	00A5D V	1
DEPTH	R	SCALR	00A32 V	1	DINE	R	SCALR	00A78 V	1	DEPT	R	SCALR	00A5D V	1
DISTKM	R	SCALR	00A81 V	1	DLAT	R	SCALR	00AC4 V	1	DISA2	R	SPRGG	EXTERN	1
DMAXH	R	SCALR	00A3B V	1	DLOT	R	SCALR	00AC5 V	1	DLEN	R	SPRGG	EXTERN	1
DMGT	R	ARRAY	00A6C V	1	DHTOR	R	SPRGG	EXTERN	1	DNAV	R	SPRGG	EXTERN	1
ELFAC	R	SCALR	0071C V	1	ELDIS	R	SCALR	00A55 V	1	ELEV	R	SCALR	00AA0 V	1
FLOAT	R	SCALR	00A54 V	1	EXIT	R	SCALR	00A55 V	1	FA	R	SCALR	00AA4 V	1
FZ	R	SPRGG	INTRIN	200	FX	R	ARRAY	0058C V	200	GFAC	R	SCALR	00A77 V	1
G1N0T	R	ARRAY	00834 V	200	GD1S	R	SCALR	00A57 V	1	HGT	R	SCALR	00A56 V	1
HT	R	SCALR	00A8C V	1	HEIGT	R	SCALR	00A57 V	1	IA	R	SCALR	00A9A V	1
IASP	R	SCALR	00A65 V	1	I	R	SCALR	00A28 V	1	IBUF	R	ARRAY	00A02 V	35
ICE	R	SCALR	00A66 V	1	IAUTH	R	SCALR	00A28 V	1	ICHG	R	SPRGG	EXTERN	1000
ICODE	R	SCALR	00A67 V	1	ICHAR	R	SCALR	00A28 V	1	IDESC	R	SCALR	00A07 V	6
ID1A8	R	SCALR	00A7D V	1	IDEP	R	SCALR	00A28 V	1	IELC	R	SCALR	00A07 V	1
IFBC	R	SCALR	00A6E V	1	IDIF	R	SCALR	00A28 V	1	IFFC	R	SCALR	00A07 V	1
IGC	R	SCALR	00A6C V	1	IFEG	R	SCALR	00A28 V	1	II	R	SCALR	00A07 V	1
II	R	SCALR	00A6C V	1	IGSUM	R	SCALR	00A28 V	1	IIOUT	R	SCALR	00A07 V	1
ILAT	R	SCALR	00A6C V	1	IIN	R	SCALR	00A28 V	1	ILONG	R	SCALR	00A07 V	1
IL00P	R	SCALR	00A6C V	1	ILM	R	SCALR	00A28 V	1	IMAX	R	SCALR	00A07 V	1
IMB	R	SCALR	00A6C V	1	IMANT	R	SCALR	00A28 V	1	IMS	R	SCALR	00A07 V	1
INCEP	R	SCALR	00A6C V	1	INENT	R	SCALR	00A28 V	1	INPT	R	SCALR	00A07 V	1
INTS	R	SCALR	00A6C V	1	IN00	R	SCALR	00A28 V	1	IOG	R	SCALR	00A07 V	1
IREGC	R	SCALR	00A6C V	1	ISEICH	R	SCALR	00A28 V	1	ISBR	R	SCALR	00A07 V	1
IS0S	R	SCALR	00A6C V	1	ISRI	R	SCALR	00A28 V	1	ISRC	R	SCALR	00A07 V	1
ISTA	R	SCALR	00A6C V	1	ISRT	R	SCALR	00A28 V	1	ISW	R	SCALR	00A07 V	1
ITSU	R	SCALR	00A6C V	1	IS2	R	SCALR	00A28 V	1	ITAPE	R	SCALR	00A07 V	1
IYIN	R	SCALR	00A6C V	1	IYRLC	R	SCALR	00A28 V	1	IZH	R	SCALR	00A07 V	1
J	R	SCALR	00A6C V	1	IYR	R	SCALR	00A28 V	1	JB	R	SCALR	00A07 V	30
JFHT	R	SCALR	00A6C V	1	JA	R	SCALR	00A28 V	1	J1	R	SCALR	00A07 V	1
J2	R	SCALR	00A6C V	1	JTAPE	R	SCALR	00A28 V	1	J4	R	SCALR	00A07 V	1
J5	R	SCALR	00A6C V	1	J3	R	SCALR	00A28 V	1	J7	R	SCALR	00A07 V	1
J8	R	SCALR	00A6C V	1	J6	R	SCALR	00A28 V	1	KDA	R	SCALR	00A07 V	1
KM	R	SCALR	00A6C V	1	K	R	SCALR	00A28 V	1	KFXN	R	SCALR	00A07 V	1
KGDA	R	SCALR	00A6C V	1	KEY	R	SCALR	00A28 V	1	KGHM	R	SCALR	00A07 V	1
KGM0	R	SCALR	00A6C V	1	KGD0	R	SCALR	00A28 V	1	KGM00	R	SCALR	00A07 V	1
KGYR	R	SCALR	00A6C V	1	KGM0	R	SCALR	00A28 V	1	KHM	R	SCALR	00A07 V	1
KN	R	SCALR	00A6C V	1	KGYR0	R	SCALR	00A28 V	1	KH0	R	SCALR	00A07 V	80
KNS	R	SCALR	00A6C V	1	KL	R	SCALR	00A28 V	1	KSH	R	SCALR	00A07 V	1
KNE	R	SCALR	00A6C V	1	KSN	R	SCALR	00A28 V	1	K1	R	SCALR	00A07 V	1
K5	R	SCALR	00A6C V	1	KYR	R	SCALR	00A28 V	1	K4	R	SCALR	00A07 V	1
K6	R	SCALR	00A6C V	1	K3	R	SCALR	00A28 V	1	K7	R	SCALR	00A07 V	1
K8	R	SCALR	00A6C V	1	K6	R	SCALR	00A28 V	1	L	R	SCALR	00A07 V	1
LABEL	R	ARRAY	003E8 V	20	K977	R	SCALR	00A28 V	1	LATH	R	SCALR	00A07 V	1
					LAT	R	SCALR	00A28 V	1					

HEX	LOC	LABEL	HEX	LOC	LABEL	HEX	LOC	LABEL	HEX	LOC	LABEL	HEX	LOC	LABEL	HEX	LOC	LABEL
001F	1		005E5	20		000F	18		0090	0		000F	1		000F	1	
00724	4		0073A	29		002F	19		0027	0		0024	2		0024	2	
00654	5		00517	60		0054	58		0054	4		0054	5		0054	5	
00639	7		00209	78		0054	75		0054	4		0054	7		0054	7	
00239	10		00209	128		0024	162		0024	0		100	100		00239	10	
00237	11		00207	138		162	174		00237	0		240	240		00237	11	
00239	12		00207	212		271	288		00239	0		302	302		00239	12	
00363	13		00207	272		304	320		00363	0		402	402		00363	13	
00468	14		00382	310		405	422		00468	0		555	555		00468	14	
00468	15		001B5	330		558	588		00468	0		588	588		00468	15	
00382	16		00233	359		588	698		00382	0		698	698		00382	16	
00405	17		00404	403		710	722		00405	0		702	702		00405	17	
00405	18		001C5	469		722	752		00405	0		752	752		00405	18	
00405	19		0048A	712		752	998		00405	0		998	998		00405	19	
00405	20		0048A	723		998	1212		00405	0		1212	1212		00405	20	
00405	21		0048A	723		1212	1502		00405	0		1502	1502		00405	21	
00405	22		0048A	723		1502			00405	0					00405	22	
00405	23		0048A	723					00405	0					00405	23	
00405	24		0048A	723					00405	0					00405	24	
00405	25		0048A	723					00405	0					00405	25	
00405	26		0048A	723					00405	0					00405	26	
00405	27		0048A	723					00405	0					00405	27	
00405	28		0048A	723					00405	0					00405	28	
00405	29		0048A	723					00405	0					00405	29	
00405	30		0048A	723					00405	0					00405	30	
00405	31		0048A	723					00405	0					00405	31	
00405	32		0048A	723					00405	0					00405	32	
00405	33		0048A	723					00405	0					00405	33	
00405	34		0048A	723					00405	0					00405	34	
00405	35		0048A	723					00405	0					00405	35	
00405	36		0048A	723					00405	0					00405	36	
00405	37		0048A	723					00405	0					00405	37	
00405	38		0048A	723					00405	0							

LOCAL VARIABLES (2805 WORDS):

0000	BLUF	003E8	LABEL	003FC	XL	0046C	CYL	004C4	GA	0058C	FX
0054	SELZ	0071C	DWGT	007E4	XM	00834	FZ	008FC	IDEG	00902	VEL
0054	THICK	00912	BG2D	008E4	JA	009E4	JB	00A02	IA	00A25	NNS
0056	FEW	00A27	FEAT	00A28	I	00A29	II	00A2A	IN	00A2B	IFUT

AN8V3  
ISW  
WHERE  
F:108  
916DATA

DISAZ  
PIN8T  
YIN8T  
MID9  
918LUSA

DMT8R  
PLET  
F:101  
MIB8  
917R

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	1896	00768
CONSTANTS:	42	0002A
LOCAL VARIABLES:	2805	00AF5
TEMPS:	1	00001
TOTAL PROGRAM:	4744	01288

C 000000 7 MAR 75

S. #N...

```

1.  G  PROGRAM NBAA
2.  C  VERSION OF 8 JUN 74 TO DB LAMONT
3.  C  VERSION OF 10 JAN 1974 TO REMOVE ABORT FOR BUFF IN ERROR
4.  C  FOR INITIAL CONVERSION OF C.AIG AND USGS 1971 IDOE
5.  C  INITIAL VERSION 10 DECEMBER 1973
6.  C  HESKANIAN G METER
7.  C
8.  DIMENSION IBUFIN(20,50,2), IBUFOT(32,50,2)
9.  DIMENSION IA(35), CRUISE(8)
10. DIMENSION IZ(9), IW(35)
11. ITAPE=1
12. JTAPE=2
13. IIN=105
14. IOUT=108
15. IREC=1
16. KK=0
17. KI=1
18. KB=2
19. IFLIP=1
20. JFLIP=1
21. KFLIP=1
22. NFLIP=1
23. ICNT=0
24. NREC=0
25. ELEV=0.
26. IBUTSW=0
27. NIN=50
28. NOUT=0
29. READ(IIN,1005) ISORC
30. WRITE(IOUT,1002) ISORC
31. IFFC=16
32. IGC=0 IGC=0
33. DEGRA=1.745329E-2
34. RADEG=57.29578
35. IENDKEY=0
36. IELC=5
37. IREGC=0
38. IFBC=0
39. C  DENS=2.67-1.03
40. DENS=1.64
41. C  OUTPUT ' PROGRAM NBAA VERS OF 8 JUN 74 '
42. C
43. C  BUFFER LOGIC FOR I/P
44. C
45. CALL BUFF IN(ITAPE,0,IBUFIN(1,1,IFLIP),1000)
46. 10 CONTINUE
47. IF (NIN.LT.50) GO TO 90
48. 15 CONTINUE
49. CALL ICHECK(ITAPE,IKEY,NI)
50. GO TO (20,50,30,40) IKEY
51. 20 OUTPUT 'WAITING FOR I/P', IEED=0
52. GO TO 15
53. 30 OUTPUT 'END OF FILE ON ITAPE', IEED=1
54. GO TO 50
55. 40 OUTPUT 'BUFF IN ERROR'
56. 50 CONTINUE
57. NIN=0
58. NFLIP=IFLIP
59. IFLIP=3-IFLIP

```

```

60.      IF (IEOD.NE.1) CALL BUFF IN(ITAPE,0,IBUFIN(1,1,IFLIP),1000)
61.      C
62.      C      INPUT LOGIC
63.      C
64.      90 CONTINUE
65.      NIN=NIN+1
66.      IF (IEOD.NE.1) GO TO 95
67.      C      GOING TO EOF PROCESSING
68.      IF (IENDKEY.EQ.1) GO TO 999
69.      NINCHK=NIN*20
70.      IF (NINCHK.GE.NI) IENDKEY=1
71.      95 CONTINUE
72.      DECODE(80,1003,IBUFIN(1,NIN,NFLIP),ND)
73.      1      (CRUISE(I),I=1,8),ITIMEZ,IYR,IMB,IDA,IHR,IMIN,
74.      2      DLAT,DLONG,NAVPT,NAVTP,ICURAZ,CURVEL,IUNCFATH,ICRRM,
75.      3      IMATHZB,IMAG,IRESMAG,IFA
76.      C
77.      C      EDIT LOGIC
78.      C
79.      IF (NAVPT.NE.0) GO TO 10
80.      IF (IFA.EQ.0) GO TO 10
81.      IF (ITIMEZ.NE.0) CALL CHGMT(IDA,IMB,IYR,KHM,KTZ,KGDA,KGMB,KGYR,
82.      1      KGHM,NTZ) ; KTZ=9 ; GO TO 100
83.      KGDA=IDA
84.      KGMB=IMB
85.      KGYR=IYR
86.      100 CONTINUE
87.      ICURVEL=IDINT(CURVEL*10.)
88.      RLAT=DLAT*DEGRA
89.      RLONG=DLONG*DEGRA
90.      THEB=GINTF(RLAT)
91.      DFA=FLOAT(IFA)*.1
92.      FA=DFA
93.      GGBS=THEB+DFA
94.      CALL BBG(K977,GBSG,GGBS,K8)
95.      IDEP=ICRRM
96.      DMIN=FLOAT(IMIN)
97.      DMINT=DMIN*.1
98.      JMIN=IDINT(DMINT)
99.      DJMIN=FLOAT(JMIN)
100.     MINT=IDINT(DMINT-DJMIN)
101.     KHM=JMIN+IHR*100
102.     KGHM=KHM
103.     DTZ=FLOAT(ITIMEZ)
104.     KTZ=IDINT(DTZ*.1)
105.     DEP=FLOAT(IDEP)
106.     BG=FA+(0.04185*DENS*DEP)
107.     IF (IDEP.EQ.0) BG=999.0
108.     PLAT=DLAT+90.;LTKEY=PLAT
109.     PLONG=DLONG+180.;LGKEY=PLONG
110.     IAKY=0
111.     ENCODE(35,1004,IZ)(CRUISE(I),I=1,8),KTZ,MINT,NAVPT,ICURAZ,
112.     1ICURVEL,IUNCFATH,IMATHZB,IMAG,IRESMAG
113.     CALL UNPKBY(IZ,IW,35)
114.     DO 120 I=1,35
115.     IA(I)=ISL(IW(I),24)
116.     120 CONTINUE
117.     C
118.     C      OUTPUT LOGIC
119.     C

```

```

120.      300 CONTINUE
121.          NOUT=NOUT+1
122.          ENCODE(128,1001,IBUF8T(1,NOUT,JFLIP),ND)IREC1,IS9RC,KGRA,KDND,
123.      1      KGYR,KGHM,DLAT,DLONG,ELEV,K977,8BSG,IDEF,FA,BG,TC,IELC,IGC,
124.      2      RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
125.      305 CONTINUE
126.          NREC=NREC+1
127.          IF(NOUT.LT.50) GO TO 10
128.      C
129.      C      BUFFER LOGIC FOR O/P
130.      C
131.      310 CONTINUE
132.          IF(IOUTSW.NE.1) IOUTSW=1,GO TO 350
133.          JKEY=ICHECK(JTAPE)
134.          GO TO (320,350,330,340) JKEY
135.      320 OUTPUT 'WAITING FOR O/P' ; IE8D=0
136.          GO TO 310
137.      330 OUTPUT 'END OF FILE JTAPE' ; IE8D=1
138.          GO TO 999
139.      340 OUTPUT 'BUFF OUT ERROR' ; IE8D=1
140.          GO TO 999
141.      350 CONTINUE
142.          NOUT=0
143.          KFLIP=JFLIP
144.          JFLIP=3-JFLIP
145.          CALL BUFF OUT(JTAPE,0,IBUF8T(1,1,KFLIP),1600)
146.          GO TO 10
147.      C
148.      C      END OF JOB
149.      C
150.      999 CONTINUE
151.      910 CONTINUE
152.          JKEY=ICHECK(JTAPE)
153.          GO TO (920,950,930,940) JKEY
154.      920 OUTPUT 'WAITING FOR O/P' ; IE8D=0
155.          GO TO 910
156.      930 OUTPUT 'BAD JKEY' ; IE8D=1
157.          GO TO 960
158.      940 OUTPUT 'BUFF OUT ERROR' ; IE8D=1
159.          GO TO 960
160.      950 CONTINUE
161.          JWDS=NOUT*50
162.          CALL BUFF OUT(JTAPE,0,IBUF8T(1,1,JFLIP),JWDS)
163.      960 CONTINUE
164.          END FILE JTAPE
165.          OUTPUT NREC
166.          OUTPUT 'ALL DONE'
167.      C
168.      C      FORMATS
169.      C
170.      1001 FORMAT(I1,I4,3I2,I4,2F9.4,F7.2,I3,F6.2,I5,2F6.1,F4.1,
171.      1      2I2,F6.1,I1,I2,35A1,1X,I1,2I3,I2)
172.      1002 FORMAT(1X,'THIS RUN PROCESSED SOURCE CODE',I5)
173.      1003 FORMAT(8A1,I5,3I2,1X,I2,I3,F8.4,F9.4,2I1,I3,F4.1,1X,2I5,I2,1X,3I5)
174.      1004 FORMAT(8A1,3I1,2I3,I5,I2,2I5)
175.      1005 FORMAT(I5)
176.      END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
SG	R	SCAL	014F3	1	BUFFIN	R	SPR3	EXTERNAL	8	BUFFOUT	R	SPR3	EXTERNAL	1
CHGMT	R	SPR3	EXTERNAL	1	CRUISE	R	SPR3	EXTERNAL	1	CURVEL	R	SPR3	EXTERNAL	1
DEGRA	R	SCAL	014EC	1	DENS	R	SCAL	014C2	1	DEP	R	SCAL	014D4	1
DFA	R	SCAL	014E6	1	DJMIN	R	SCAL	014EF	1	DLAT	R	SCAL	014CF	1
CLONG	R	SCAL	014D0	1	ELEV	R	SCAL	014EC	1	DMINT	R	SCAL	014ED	1
DTZ	R	SCAL	014F1	1	GINTF	R	SPR3	EXTERNAL	1	FA	R	SCAL	014E7	1
FL0AT	R	SPR3	EXTERNAL	1	IA	R	SPR3	EXTERNAL	1	G885	R	SCAL	014E8	1
IBUFIN	R	SCAL	014C8	1	IBUF0T	R	SPR3	EXTERNAL	1	IAKEY	R	SCAL	014F8	1
ICNT	R	SCAL	00000	2000	ICORRM	R	SCAL	007D0	35	ICHECK	R	SPR3	EXTERNAL	1
ICURVEL	R	SCAL	014B3	1	IZ	R	SCAL	014D6	1	ICURAZ	R	SCAL	014D3	1
IDINT	R	SCAL	014E2	1	IELC	R	SCAL	014CC	1	IDEP	R	SCAL	014E9	1
IE8D	R	SCAL	014C5	1	IFA	R	SCAL	014BF	1	IENDKEY	R	SCAL	014BE	1
IFFC	R	SCAL	014BA	1	IFLIP	R	SCAL	014DA	1	IFBC	R	SCAL	014C1	1
IM8	R	SCAL	014CD	1	IIN	R	SCAL	014AF	1	IGK	R	SCAL	014B5	1
IMAG	R	SCAL	014D8	1	IMATH28	R	SCAL	014A9	1	IMIN	R	SCAL	014C3	1
IM8	R	SCAL	014CR	1	IBUT	R	SCAL	014D7	1	IOUTSM	R	SCAL	014E6	1
IIEC1	R	SCAL	014AB	1	IS8RC	R	SCAL	014AA	1	IRESMAG	R	SCAL	014D9	1
ISL	R	SCAL	014C9	1	IUNCFATH	R	SCAL	014B9	1	ITAPE	R	SCAL	014A7	1
ITIMEZ	R	SCAL	014CA	1	I2	R	SCAL	014D5	1	IW	R	SCAL	01484	1
JKEY	R	SCAL	014FB	1	JMIN	R	SCAL	014EE	1	JFLIP	R	SCAL	014B0	1
JMS	R	SCAL	014FC	1	KFLIP	R	SCAL	014B1	1	JTAPE	R	SCAL	014A8	1
KGHM	R	SCAL	014E0	1	KGM8	R	SCAL	014DE	1	KDA	R	SCAL	014D0	1
KHM	R	SCAL	014DB	1	KI	R	SCAL	014AD	1	KGYR	R	SCAL	014DF	1
K8	R	SCAL	014AE	1	KZ	R	SCAL	014DC	1	KK	R	SCAL	014AC	1
LGKEY	R	SCAL	014F7	1	LTKEY	R	SCAL	014F5	1	K977	R	SCAL	014E9	1
NAVPT	R	SCAL	014D1	1	NAVTYPE	R	SCAL	014D2	1	MINT	R	SCAL	014FO	1
NFLIP	R	SCAL	014B2	1	NI	R	SCAL	014C4	1	ND	R	SCAL	014C7	1
NINCHK	R	SCAL	014C6	1	NBUT	R	SCAL	014B8	1	NREC	R	SCAL	014B7	1
NTZ	R	SCAL	014E1	1	88G	R	SPR3	EXTERNAL	1	88SG	R	SCAL	014B4	1
PLAT	R	SCAL	014F4	1	PLONG	R	SCAL	014F6	1	RADEG	R	SCAL	014EA	1
RFA	R	SCAL	014FA	1	RLAT	R	SCAL	014E3	1	RL8G	R	SCAL	014BD	1
TC	R	SCAL	014F9	1	THE8	R	SCAL	014E5	1	UNPKBY	R	SPR3	EXTERNAL	1

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
10	00058	15	00058	20	00069	30	00075	40	00083
90	000A0	95	000AF	100	000EF	120	00167	300	00169
310	00134	320	001A7	330	001B3	340	001C0	350	001CC
920	001E8	930	001F7	940	00202	950	0020E	960	0021B
1001	0022E	1002	00243	1003	0024E	1004	0025E	1005	00265

LOCAL VARIABLES (5373 WORDS):

00000	IBUFIN	00700	IBUF0T	01473	CRUISE	01484	IW
01447	ITAPE	014A3	JTAPE	014AA	IBUT	014AC	KK
01440	KI	014AE	K9	014B0	JFLIP	014B2	NFLIP
01433	ICNT	014B4	NREC	014B6	IBUTSM	014B8	NBUT
014B9	IS8RC	014BA	IFFC	014B8	DEGRA	014BE	TENDKEY
014BF	IELC	014C3	IIEC1	014C3	DENS	014C4	NI
014C5	IE8D	014C6	NINCHK	014C8	I	014CA	IYR
014CB	IM8	014CC	IDA	014CE	IMIN	014CD	DL8G

014D1 NAVPT	014D2 NAVTYP	014D3 ICURAZ	014D4 CURVEL	014D5 IUNCFATH	014D6 IC9RRM
014D7 IMATH28	014D8 IMAG	014D9 IRESMAG	014DA IFA	014DB KHM	014DC KTZ
014DD KGDA	014DE KGM8	014DF KGYR	014EO KGHM	014E1 NTZ	014E2 ICJRVEL
014E3 RLAT	014E4 RL8NG	014E5 THE8	014E6 DFA	014E7 FA	014E8 G88S
014E9 K977	014EA 88SG	014EB IDEP	014EC DMIN	014ED DMINT	014EE JMIN
014EF DJMIN	014FO MINT	014F1 DTZ	014F2 DEP	014F3 BG	014F4 PLAT
014F5 LTKEY	014F6 PL8NG	014F7 LGKEY	014F8 IAKY	014F9 TC	014FA RFA
014FB JKEY	014FC JWDS				

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

FL8AT	IDINT	ISL
-------	-------	-----

EXTERNAL SUBPROGRAMS REQUIRED:

BUFFIN	BUFF8UT	CHGMT	GINTF	ICHECK	8BG	UNPKBY	F:101
F:102	F:103	F:104	F:105	F:106	F:108	M:DB	M:8C
9BCDREAD	9BCDWRIT	9DEC8DE	9ENC8DE	9ENDFILE	9ENDI8L	9INITIAL	9I8DATA
9I8LUSA	9IT8R	9PRINT	9RT8I	9ST8P			

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	617	DEC	HEX
CONSTANTS:	13	WORDS	
LOCAL VARIABLES:	5373		
TEMPS:	1		
TOTAL PROGRAM:	6004		
			01774

```

1.  C      PROGRAM PR0FG
2.  C
3.      DIMENSION PLT(15)
4.      DIMENSION IA(35)
5.  C
6.  C      VERSION OF 2 OCT 75 TO CALL GINBT
7.  C      PROGRAM PR0FG, PLOTS PROFILE OF SELECTED VARIABLE
8.  C      READ FROM GSUM FORMAT
9.  C
10. C
11. C      X AXIS PROPORTIONAL TO EITHER DISTANCE OR TIME ALONG TRACK
12. C
13. C      IF LCNT=9000 OR GREATER THEN TIME ANOTATION
14. C      IS DELETED
15. C
16. C      HAS DATA AND DISTANCE LIMITS AS INPUT
17. C
18. C      FOR SELECTION OF SINGLE STATION SSW(5) = 1 AND MSTA READ IN
19. C      AS THE STATION NUMBER TO BE SELECTED
20. C      USES GINBT, FIND, ISW, STAT, EVIL, CDATE, MCVOL, SPOT,
21. C
22. C      SSW(5)=1 TO SELECT SINGLE STATION NUMBER
23. C      SSW (14) = UP TO CALL FIND
24. C
25. C
26. C      DIMENSION IBUF(1000)
27. C
28. C
29. C      IIN = 105
30. C      IIBUT = 108
31. C      KK=0
32. C      JTAPE=2
33. C      KI=1
34. C      INNT=ISW(=2)
35. C      CALL GINBT(ITAPE,JTAPE,KK)
36. C      CALL PLOTS (IBUF, =1000)
37. C      WRITE (IIBUT,600)
38. C      600 FORMAT(/,'PROGRAM PR0FG VER 2 OCT 75')
39. C
40. C      *****
41. C      ITAPE = INPUT TAPE
42. C      ITAPE = 1
43. C      *****
44. C
45. C      NEBF = 0
46. C      XX=0.0
47. C      INIT=1
48. C      DIFAC=NUMBER OF N MILES,KM,OF HOURS PER INCH ON PLOT
49. C      YFAC = ENGINEERING UNITS PER INCH ON PLOT FOR Y DIRECTION
50. C      LCNT = TIME ANOTATION EVERY LCNT POINTS PLOTTED
51. C      MIKM = 0 FOR NAUTICAL MILES, 1 FOR KILOMETERS
52. C      NFILE = NO. OF INPUT REELS TO PROCESS
53. C      READ (IIN,4) DIFAC, YFAC, LCNT, MIKM, NPL0T, NFILE
54. C      4  FORMAT (2F10.0, 4I5)
55. C      OUTPUT DIFAC, YFAC, LCNT, MIKM, NPL0T, NFILE
56. C      ULIM=UPPER LIMIT FOR PLOTTING DATA VALUE IN ENG UNITS
57. C      BLIM=BOTTOM LIMIT FOR PLOTTING DATA VALUE IN ENG UNITS
58. C      DLIM = DISTANCE LIMIT IN INCHES FOR PLOTTING DATA POINT
59. C      IXDIR=1 FOR X AXIS PROPORTIONAL TO DISTANCE ALONG TRACK

```

```

60. C      =2 FOR X AXIS PROPORTIONAL TO TIME ALONG TRACK
61.      READ (IIN,6) ULM, BLIM, DLIM, IXDIR
62.      6      FORMAT (3F10.0, 15)
63. C      XALW = ALLOWABLE INCHES FOR LENGTH OF PLOT
64. C      BEFORE REINITIALIZATION
65. C      DMOVE = INCHES TO BE SPACED BEFORE REINITIALIZATION
66.      READ (IIN,8) XALW, DMOVE
67.      8      FORMAT (2F10.0)
68.      OUTPUT ULM, BLIM, DLIM, IXDIR, XALW, DMOVE
69.      IF (ISW(14)) 129, 129, 128
70.      128     READ (IIN,9) LIMDA, LIMMB, LIMYR, LIMHM
71.      9      FORMAT (6I5)
72.      OUTPUT LIMDA, LIMMB, LIMYR, LIMHM
73.      111     CALL FIND (LIMDA, LIMMB, LIMYR, LIMHM, KGDA, KGMB, KGYR, KGHM, INDK)
74.      IF (INDK) 111, 129, 129
75.      129     CONTINUE
76.      READ (IIN,9) MSTA
77.      CALL WHERE (XORG, YORG, RFACT)
78.      CALL PLOT(XORG, YORG, -3)
79.      IF (8990-LCNT) 130, 132, 132
80.      130     NCNT=0
81.      GO TO 134
82.      132     NCNT=LCNT
83.      134     IF (MIKM) 14, 13, 14
84.      13      CNV=0.53959
85.      GO TO 52
86.      14      CNV=1.0
87.      50      CONTINUE
88.      15      CONTINUE
89.      CALL GINBT (ITAPE, JTAPE, KI, KGDA, KGMB, KGYR, KGHM, IDIF, ISORC,
90.      *      RLAT, RLONG, ELEV, K977, GBSG, IDEP, FA, BG, TC, IELC, IGC, RFA, IREGC,
91.      *      IFFC, IA, IFBC)
92.      52      CONTINUE
93.      IF (KI.EQ.9) GO TO 44
94.      IF (ISW(5).NE.1) GO TO 70
95.      IF (KGHM.EQ.MSTA) GO TO 70
96.      GO TO 52
97.      44      OUTPUT 'PLOTING COMPLETED'
98.      CALL PLOT (XX, YY, 999)
99.      CALL EXIT
100.      70      DEPTH=IDEP
101.      IF (IDEP) 78, 74, 78
102.      74      HEIGHT=ELEV
103.      GO TO 80
104.      78      HEIGHT = -DEPTH
105.      80      A=K977.977
106.      A=A*1000.0
107.      GBS=GBSG+A
108.      BGC=M+BG+TC
109.      100     PLT(1)=KGHM
110.      PLT(2)=ISORC
111.      PLT(3)=ELEV
112.      PLT(4)=DEPTH
113.      PLT(5)=HEIGHT
114.      PLT(6)=FA
115.      PLT(7)=BG
116.      PLT(8)=TC
117.      PLT(9)=BGC
118.      PLT(10)=RFA
119.      PLT(11)=GBS

```

```

120.      KDA=KGDA
121.      KM0=KGM0
122.      KYR=KGYR
123.      KHM=KGHM
124.      DAY=KGDA
125.      YM0=KGM0
126.      YEAR=KGYR
127.      HOUR=KGHM
128.      XLAT=RLAT
129.      YL0NG=RL0NG
130.      DATA=PLT(NPL0T)
131.      IF (INIT-1)25,30,25
132. 30    DISTM=0.0
133.      TIMD=0.0
134.      KDA0=KGDA
135.      KM00=KGM0
136.      KYR0=KGYR
137.      KHM0=KGHM
138.      INIT=0
139.      IPEN=3
140.      GO TO 50
141. 25    TLAT=ABS(XLAT)
142.      RAD1=6371229.0
143.      DLTDI=(XLAT-XLAT0)*RAD1
144.      DLGDI=(XL0NG-XLNG0)*RAD1*C0S(TLAT)
145.      DISTM=SQRT((ABS(DLTDI))**2+(ABS(DLGDI))**2)
146.      CALL CDATE(KDA0,KM00,KYR0,KHM0,KDA,KM0,KYR,KHM,TIMD)
147. 350   XLAT0=XLAT
148.      XLNG0=XL0NG
149.      KDA0=KDA
150.      KM00=KM0
151.      KYR0=KYR
152.      KHM0=KHM
153.      YY=DATA/YFAC
154.      IF (IXDIR=1)54,56,54
155. 54    DIS=TIMD/DIFAC
156.      GO TO 58
157. 56    DIS=(DISTM*0.001*C0NV)/DIFAC
158. 58    IF (DIS-DLIM)365,365,360
159. 360   XX=XX+3.0
160.      WRITE (IIBUT,61)KDA,KM0,KYR,KHM
161. 61    FORMAT('DLIM',3I3,15)
162.      CALL PL0T(XX,0.0,3)
163.      GO TO 25
164. 365   IF (DATA=ULIM)368,366,366
165. 366   WRITE (IIBUT,67)KDA,KM0,KYR,KHM
166. 67    FORMAT('ULIM',3I3,15)
167.      GO TO 160
168. 167   WRITE (IIBUT,168)KDA,KM0,KYR,KHM
169. 168   FORMAT('BLIM',3I3,15)
170. 160   XX=XX+DIS
171.      CALL PL0T(XX,0.0,3)
172.      IPEN=3
173.      GO TO 15
174. 368   IF (BLIM=DATA)369,167,167
175. 369   XX=XX+DIS
176. C     CHECKING WITHIN ALLOWABLE PL0T DISTANCE
177. 170   IF (XX=XAL0W)180,172,172
178. 172   XX=XX+DM0VE
179.      IPEN=3

```

```

180.      CALL PLOT(XX,YY,IPEN)
181.      CALL WHERE (XORG, YORG, RFACT)
182.      CALL PLOT(XORG,0.0,-3)
183.      XX=0.0
184.      CALL PLOT(XX,YY,IPEN)
185.      CALL SPOT(XX,YY)
186.      IPEN=2
187.      GO TO 72
188.      180 CALL PLOT(XX,YY,IPEN)
189.      CALL SPOT(XX,YY)
190.      IPEN=2
191.      72 CONTINUE
192.      IF(NCNT-LCNT)120,300,120
193.      300 CALL NUMBER (XX, =5.0, 0.07, DAY, 90.0, =1)
194.      CALL NUMBER (XX, =4.8, 0.07, YMO, 90.0, =1)
195.      CALL NUMBER (XX, =4.6, 0.07, YEAR, 90.0, =1)
196.      CALL NUMBER (XX, =4.4, 0.07, HOUR, 90.0, =1)
197.      CALL PLOT(XX,YY,3)
198.      NCNT=1
199.      GO TO 15
200.      120 NCNT=NCNT+1
201.      OUTPUT NCNT
202.      GO TO 15
203.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	00453 V	1	ABS	R	SPR8G	INTRIN	1	BG	R	SCALR	00448 V	1
BGMH	R	SCALR	00455 V	1	BLIM	R	SCALR	00428 V	1	CDAE	R	SPR8G	EXTERN	1
CRNV	R	SCALR	00456 V	1	CBS	R	SPR8G	INTRIN	1	DATA	R	SCALR	00460 V	1
DAY	R	SCALR	00457 V	1	DEPTM	R	SCALR	00451 V	1	DIFAC	R	SCALR	00424 V	1
DIS	R	SCALR	00458 V	1	DLTDM	R	SCALR	00452 V	1	DLGDI	R	SCALR	0046C	1
DLIM	R	SCALR	00459 V	1	DLTDI	R	SCALR	0046A V	1	DM8VE	R	SCALR	0042F V	1
ELEV	R	SCALR	00460 V	1	EXIT	R	SPR8G	EXTERN	1	FA	R	SCALR	00447 V	1
FIND	R	SCALR	00461 V	1	GINBT	R	SPR8G	EXTERN	1	GBBS	R	SCALR	00454 V	1
HEIGT	R	SCALR	00462 V	1	H0UK	R	SCALR	0045D V	1	IA	R	ARRAY	0000F V	35
I0UF	R	SCALR	00463 V	1000	IDFH	R	SCALR	00446 V	1	IDIF	R	SCALR	0043F V	1
IELC	R	SCALR	00464 V	1	IFBC	R	SCALR	0044F V	1	IFFC	R	SCALR	0044E V	1
IGC	R	SCALR	00465 V	1	IIN	R	SCALR	0044A V	1	IIBUT	R	SCALR	0041B V	1
INOK	R	SCALR	00466 V	1	INIT	R	SCALR	00423 V	1	INNT	R	SCALR	0041F V	1
IPEN	R	SCALR	00467 V	1	IRE9C	R	SCALR	0044D V	1	IS8RC	R	SCALR	00440 V	1
ISM	R	SPR8G	EXTERN	1	ITAPE	R	SCALR	00420 V	1	IXDIR	R	SCALR	0042D V	1
JTAPE	R	SCALR	00468 V	1	KDA	R	SCALR	00456 V	1	KDA9	R	SCALR	00463 V	1
KGOA	R	SCALR	00469 V	1	KGMH	R	SCALR	00437 V	1	KDM9	R	SCALR	00435 V	1
KGYR	R	SCALR	00470 V	1	KHM	R	SCALR	00459 V	1	KHM9	R	SCALR	00466 V	1
KI	R	SCALR	00471 V	1	KYR	R	SCALR	0041C V	1	KMB	R	SCALR	00457 V	1
KMB9	R	SCALR	00472 V	1	KYR	R	SCALR	00458 V	1	KYR9	R	SCALR	00465 V	1
K977	R	SCALR	00473 V	1	LCNT	R	SCALR	00426 V	1	LIMDA	R	SCALR	00430 V	1
LIMHM	R	SCALR	00474 V	1	LIMM0	R	SCALR	00431 V	1	LIMYR	R	SCALR	00432 V	1
MIKM	R	SCALR	00475 V	1	MSTA	R	SCALR	00439 V	1	NCNT	R	SCALR	0043D V	1
NE9F	R	SCALR	00476 V	1	NFILE	R	SCALR	00429 V	1	NPL9T	R	SCALR	00428 V	1
NUMBER	R	SPR8G	EXTERN	1	BS9J	R	SCALR	00445 V	1	PL9T	R	SPR8G	EXTERN	1
PL9TS	R	SPR8G	EXTERN	1	PLT	R	ARRAY	00000 V	15	RADI	R	SCALR	00469 V	1
RFA	R	SCALR	00477 V	1	RFAC	R	SCALR	0043C V	1	RLAT	R	SCALR	00441 V	1
RL9NG	R	SCALR	00478 V	1	SP9T	R	SPR8G	EXTERN	1	SGRT	R	SPR8G	INTRIN	1
TC	R	SCALR	00479 V	1	TMD	R	SCALR	00462 V	1	TLAT	R	SCALR	00468 V	1
ULIM	R	SCALR	00480 V	1	WHERE	R	SPR8G	EXTERN	1	XAL9W	R	SCALR	0042E V	1
XLAT	R	SCALR	00481 V	1	XLATB	R	SCALR	0046B V	1	XLNG9	R	SCALR	0046E V	1
XL9NG	R	SCALR	00482 V	1	XMRG	R	SCALR	0043A V	1	XX	R	SCALR	00422 V	1
YEAR	R	SCALR	00483 V	1	YFAC	R	SCALR	00425 V	1	YLBNG	R	SCALR	0045F V	1
YMR	R	SCALR	00484 V	1	YMRG	R	SCALR	0043B V	1	YY	R	SCALR	00450 V	1

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
4	00037	6	00062	8	0006C	9	0009C	13	000DB	14	000DE	14	000DE	14	000DE
15	000E0	25	00177	30	00166	44	00107	50	000E0	52	000F8	52	000F8	52	000F8
54	001AC	56	00180	58	001B5	61	001C3	67	001D9	70	00118	70	00118	70	00118
72	0022D	78	0011D	78	00120	80	00123	100	00130	111	000B5	111	000B5	111	000B5
120	00258	128	00094	129	000C2	130	000D4	132	000D7	134	000D9	134	000D9	134	000D9
150	001EC	167	001DF	168	001E7	170	001FD	172	00200	180	00222	180	00222	180	00222
300	00270	350	00194	360	001B8	365	001CE	366	001D1	368	001F7	368	001F7	368	001F7
369	001FA	600	0001C												

LOCAL VARIABLES (1136 WORDS):

00000	PLT	0000F	1A	00032	IRUF	0041A	IIN	0041B	I19JT	0041C	KK
00410	JTAPE	0041E	KI	0041F	IANN7	00420	ITAPF	00421	NEBF	00422	XX
00423	INIT	00424	DIFAC	00425	YFAC	00426	LCNT	00427	YIKM	00428	NPL9T
00429	NFILE	0042A	ULIM	0042B	BLIM	0042C	DLIM	0042D	IXDIR	0042E	XAL9W
0042F	DM8VE	00430	LIMDA	00431	LIMM9	00432	LIMYR	00433	LIMHM	00434	KGDA

00435 KGM0	00436 KGYR	00437 KGHM	00438 INDK	00439 MSTA	0043A X0RG
0043B YRRG	0043C RFACT	0043D NCNT	0043E C0NV	0043F IDIF	00440 IS0RC
00441 RLAT	00442 RLNG	00443 ELEV	00444 K977	00445 0BSG	00446 IDEP
00447 FA	00448 BG	00449 TC	0044A IELC	0044B IGC	0044C RFA
0044D IREGC	0044E IFFC	0044F IFBC	00450 YY	00451 DEPTH	00452 HEIGHT
00453 A	00454 GRS	00455 BGC0M	00456 KDA	00457 KM0	00458 KYR
00459 KHM	0045A DAY	0045B YM0	0045C YEAR	0045D H0UR	0045E XLAT
0045F YLNG	00460 DATA	00461 DISTM	00462 TIMD	00463 KDA8	00464 KM00
00465 KYR0	00466 KHM0	00467 IPEN	00468 TLAT	00469 RAD1	0046A DLTD1
0046B XLAT0	0046C DLGDI	0046D XLNG	0046E XLNG0	0046F DIS	

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

ABS	C0S	SQRT
-----	-----	------

EXTERNAL SUBPROGRAMS REQUIRED:

COATE	EXIT	FIND	GIN0T	ISM	NUMBER	PL0T	PL0TS
SP0T	WHERE	F:101	F:102	F:103	F:104	F:105	F:106
F:108	M:00	M:0C	9BCDREAD	9BCDWRIT	9C9S	9ENDI0L	9INITIAL
9I0DATA	9I0R	9PRINT	9SQRT				

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC	HEX
GENERATED CODE:	611	00263
CONSTANTS:	21	00015
LOCAL VARIABLES:	1136	00470
TEMPS:	1	00001
TOTAL PROGRAM:	1769	006E9

(NO MEMORY PROTECTION)

COMPILED 6 MAY 74

```

1.  C      PROGRAM PROJ4
2.  C
3.  C      VERSION 6 MAY 74, CHANGE USE OF SIN02 TO SIN0T
4.  C      VERSION OF 4 FEB 1973, TO OUTPUT DECIMAL DEGREES IN 1A FIELD
5.  C      VERSION OF 25 AUGUST 1972, TO UPDATE YIN0T CALLING ARGUMENTS
6.  C      VERSION OF 14 MARCH 1972, ADDING IMANT TO ARG LIST TO PIN0T
7.  C      VERSION OF 29 AUGUST 1971
8.  C
9.  C      PROGRAM PROJ4, PROJECTS SEVERAL DATA FORMATS TO A STRAIGHT LINE
10. C      CAN PROCESS DIFFERENT FORMATS IN SAME RUN
11. C      ON FORMAT CHOICE IN PUT CARD, PUT JFMTS IN ORDER DESIRED FOR
12. C      PROCESSING DURING RUN
13. C
14. C      JFMT ALLWS CHOICE OF DATA FORMAT TO BE PROCESSED
15. C      JFMT = 1 FOR GSUM FORMAT
16. C      = 2 FOR SEAG1 INPUT FORMAT, OUTPUT IS GSUM FORMAT
17. C      = 3 FOR SPFMT FORMAT
18. C      = 4 FOR SEISMICITY DATA FORMAT
19. C      = 7 FOR ACTIVE VOLCANOES
20. C
21. C
22. C      ITAPE IS SET TO (JFMT + 2)
23. C      JTAPE = 2
24. C      KTAPE=20
25. C
26. C
27. C
28. C      SHOULD HAVE ASSIGN CARDS FOR UNITS 2,3,4,5,6,9,20
29. C
30. C      UNIT 2 IS FOR OUTPUT OF PROJECTED DATA
31. C      UNIT 3 FOR GSUM
32. C      UNIT 4 FOR SEAG1
33. C      UNIT 5 FOR SPFMT DATA
34. C      UNIT 6 FOR SEISMICITY DATA
35. C      UNIT 9 FOR ACTIVE VOLCANOE DATA
36. C      UNIT 20 FOR DISK STORAGE OF GSUM DATA IF ISW(3)=1
37. C
38. C
39. C
40. C      TO STOP PROCESSING MAKE START DAY=99
41. C      IF START DAY = 99, PROGRAM GOES TO 1000 AND CHECKS INPUT FORMAT
42. C      LABELS FOR FORMAT CODE NUMBER OF ZERO
43. C
44. C
45. C      SSW(3) = 1 TO OUTPUT GSUM DATA ONTO A DISK FILE, ONLY
46. C      APPLIES WHEN JFMT = 1 OR 2
47. C
48. C      DIMENSION IA(35),JF(10)
49. C      DIMENSION IDESC(6),VEL(8),THICK(8),X(8)
50. C      DIMENSION N0W(4)
51. C      DIMENSION IZ( 9),IW(35)
52. C      CALL STAT

```

```

53.      NWGN=1
54.      IIN=105
55.      IIBUT=108
56.      JTAPE=2
57.      JFCT=1
58.      JJREC=0
59.      I8=8
60.      IREC1=1
61.      DEGRA=1.745329E-02
62.      RADEG=57.29578
63.      C PRINT DATE AND TIME OF JOB ON HEADING
64.      CALL T8DAY(N8W)
65.      WRITE(IIBUT,13) N8W
66.      13 FORMAT(1X,4A4)
67.      WRITE(IIBUT,16)
68.      16 FORMAT('PROJ4 RUN, VERSION OF 6 MAY 1974')
69.      C
70.      C INITIALIZING PROJECTION REQUIREMENTS
71.      C
72.      C KK=0
73.      CALL PROJ(KK,RLAT,RLONG,PLAT,PLONG,DISR,DIST)
74.      C
75.      C
76.      C READING ORDER THAT DATA FORMATS ARE TO BE PROCESSED
77.      C
78.      READ(IIN,20)JFMT,JF(2),JF(3),JF(4),JF(5),JF(6),JF(7),JF(8),JF(9)
79.      20 FORMAT(9I5)
80.      C
81.      C
82.      C START OF READING NEW DATA FORMAT BEGINS HERE WITH SSW CARD
83.      C
84.      18 INIT=ISW(-2)
85.      CALL SETSKP(IND)
86.      WRITE(IIBUT,25) JFMT
87.      25 FORMAT('JFMT =',I4)
88.      ITAPE=JFMT+2
89.      GO TO (40,50,80,90,100,1000,110)JFMT
90.      40 KK=0
91.      CALL SIN8T(ITAPE,JTAPE,KK,KGDA,KGM8,
92.      1 KGYR,KGHM,IDIF,IS8RC,RLAT,RLONG,ELEV,K977,8BSG,
93.      2 IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
94.      GO TO 100
95.      50 READ(IIN,55)IS8RC,IELC,IGC
96.      55 FORMAT(3I5)
97.      ELEV=0.0
98.      I8GR=88
99.      TC=99.9
100.     RFA=0.0
101.     IREGC=0.0
102.     IFFC=0.0
103.     IFBC=0
104.     KK=0
105.     CALL SIN8T(ITAPE,JTAPE,KK,KGDA,KGM8,KGYR,KGHM,IDIF,RLAT,

```

```

106.      1  RLONG,KVN,KVE,K977,IGR,KFA,KBG,KCVN,KCVE,
107.      2  KCDM,MTDC,MT,MAG1,MAG2,KETV8)
108.      CALL      GIN8T(ITAPE,UTAPE,KK,KGDA,KGM8,
109.      1  KGYR,KGHM,IDIF,IS8RC,RLAT,RLONG,ELEV,K977,8BSG,
110.      2  IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
111.      GO TO 100
112.  80  CONTINUE
113.      KK=0
114.      CALL      PIN8T(ITAPE,UTAPE,KK,ISTA,KEY,LAT,LATM,KNS,
115.      1  LONG,L9M,KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,
116.      2  DINE,STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTW)
117.      GO TO 100
118.  90  CONTINUE
119.      KK=0
120.      CALL      YIN8T(ITAPE,UTAPE,KK,
121.      1  ISR1,ISR2,KDA,KM8,KYR,KHM,SEC,DLAT,KSND,DL8N,KWE,DEPT,AMAG,IMB,
122.      2  IS8S,INTS,IDIAS,ITSU,ISEICH,IV8LC,IN8NT,IWG,IFEG,IMS,IASP,IZH,
123.      3  ICE,IMG,IAUTH,IQHY,NPP,I8Q,ILM,IS1,IS2)
124.      GO TO 100
125.  C   FOR PRESENT THERE IS NO VOLCANOE STREAM
126.  110 GO TO 1000
127.  100 CONTINUE
128.  C   READING START & END DATE & ISKP CARD FOR EACH INPUT FORMAT
129.      IFLAG=0
130.      READ(IIN,2)ISTDA,ISTM8,ISTYR,ISTHM,IENDA,IENM8,IENYR,IENHM,ISKP
131.  2   FORMAT(      3I2,I4,5X,3I2,I4,5X,I5)
132.      WRITE(IIBUT,6365)ISTDA,ISTM8,ISTYR,ISTHM,IENDA,IENM8,IENYR,
133.      1 IENHM,ISKP
134.  6365 FORMAT('PR0J4:  START DATE ',3I2,I4,',  END DATE ',3I2,I4,',  ISKP=
135.      1',I4)
136.      IF (ISKP.EQ.0) GO TO 8
137.      CALL SKPREC(ITAPE,ISKP)
138.      GO TO (999,8,999,999,999,999) IND
139.      8  CONTINUE
140.  C   CHECK IF END OF PROCESSING
141.  C   BY CHECK IF ISTDA=99
142.      IF (ISTDA.EQ.99) GO TO 992
143.  C
144.  C   INITIALIZATION NOW COMPLETE
145.  C
146.  C   READ IN DATA
147.  C
148.  150 GO TO (200,300,400,500,1000,1000,530)JFMT
149.  200 KK=1
150.      CALL      GIN8T(ITAPE,UTAPE,KK,KGDA,KGM8,
151.      1  KGYR,KGHM,IDIF,IS8RC,RLAT,RLONG,ELEV,K977,8BSG,
152.      2  IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
153.      IF (KK=9)210,1000,210
154.  210 GO TO 186
155.  300 KK=1
156.      CALL      SIN8T(ITAPE,UTAPE,KK,KGDA,KGM8,KGYR,KGHM,IDIF,RLAT,
157.      1  RLONG,KVN,KVE,K977,IGR,KFA,KBG,KCVN,KCVE,
158.      2  KCDM,MTDC,MT,MAG1,MAG2,KETV8)

```

```

159.      IF(KK-9)310,1000,310
160.      310 DBSG=I0GR
161.      DBSG=DSBG*0.1
162.      FA=FL0AT(KFA)*0.1
163.      BG=FL0AT(KBG)*0.1
164.      GO TO 186
165.      400 KK=1
166.      CALL      PIN0T(ITAPE,UTAPE,KK,ISTA,KEY,LAT,LATM,KNS,
167.      1  LONG,L0M,KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,
168.      2  DINE,STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTW)
169.      IF(KK-9)410,1000,410
170.      410 RLATM=LATM
171.      RL0M=L0M
172.      CALL NAVIN(LAT,RLATM,KNS,LONG,RL0M,KEW,RLAT,RLONG)
173.      GO TO 186
174.      500 KK=1
175.      CALL      YIN0T(ITAPE,UTAPE,KK,
176.      1  ISR1,ISR2,KDA,KM0,KYR,KHM,SEC,DLAT,KSN,DL0N,KWE,DEPT,AMAG,IMB,
177.      2  IS0S,INTS,IDIAS,ITSU,ISEICH,IV0LC,IN0NT,IWG,IFEG,IMS,IASP,IZH,
178.      3  ICE,IMG,IAUTH,IGHY,NPP,I0Q,ILM,IS1,IS2)
179.      IF(KK-9)510,1000,510
180.      510 KL=0
181.      CALL      DNAV(DLAT,KSN,DL0N,KWE,RLAT,RLONG,KL)
182.      GO TO 186
183.      530 GO TO 1000
184.      186 CONTINUE
185.      IF(KK-9)190,992,190
186.      C CHECKING FOR ERR ON INPUT DATA
187.      190 JDA=KGDA
188.      JMB=KGM0
189.      JYR=KGYR
190.      JHM=KGHM
191.      C IFLAG IS A FLAG TO ALLOW SKIPPING THE FIRST CALL TO FIND
192.      C IF WE HAVE ALREADY FOUND THE STARTING DATE
193.      C
194.      IF(IFLAG.NE.0) GO TO 82
195.      CALL FIND(ISTDA,ISTM0,ISTYR,ISTHM,JDA,JMB,JYR,JHM,INDK)
196.      IF(INDK.EQ.-1) GO TO 150
197.      IFLAG=1
198.      82 CONTINUE
199.      IF(IENYR.EQ.0) GO TO 851
200.      CALL FIND(IENDA,IENM0,IENYR,IENHM,JDA,JMB,JYR,JHM,INDK)
201.      IF(INDK.EQ.1) GO TO 995
202.      851 CONTINUE
203.      852 CONTINUE
204.      C
205.      C
206.      C START PROCESSING
207.      C
208.      C
209.      KK=NW0N
210.      CALL      PROJ(KK,RLAT,RLONG,PLAT,PLONG,DISR,DIST)
211.      C CHECKING IF DATA POINT IS WITHIN AREA AND DISTANCE FROM PROJ LINE

```

```

212.      IF(KK+2)599,150,599
213.      599 CONTINUE
214.      JJREC=JJREC+1
215.      C      X=DISR*6371.0
216.      C      BUTPUT X
217.      C
218.      C      BUTPUT DATA
219.      C
220.      GO TO (600,600,700,800,1000,1000,830)JFMT
221.      600 KK=-2
222.      C
223.      C      LOGIC OFR PUTTING RLAT AND RLONG VALUES IN ARRAY 'IA'
224.      C
225.      C      GROUPING VARIABLES FOR BUTPUT UNDER ARRAY IA
226.      DLAT=PLAT*RADEG
227.      DLONG=RLONG*RADEG
228.      ENCODE(35,608,IZ) DLAT,DLONG
229.      608 FORMAT(2F9.4,17X)
230.      CALL UNPKBY(IZ,IW,35)
231.      DO 609 J=1,35
232.      IA(J)=ISL(IW(J),24)
233.      609 CONTINUE
234.      C
235.      C      PUTS PLAT AND LONG IN POSITION OF RLAT AND RLONG
236.      C
237.      CALL GINBT(ITAPE,JTAPE,KK,KGDA,KGMB,
238.      1 KGYR,KGHM,IDIF,ISORC,PLAT,PLONG,ELEV,K977,OBSEG,
239.      2 IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
240.      IF(ISW(3))900,900,612
241.      612 WRITE(KTAPE,12)IREC1,ISORC,KGDA,KGMB,KGYR,KGHM,
242.      1 PLAT,PLONG,ELEV,K977,OBSEG,IDEP,FA,BG,TC,IELC,IGC,
243.      2 RFA,IREGC,IFFC,IA,IFBC
244.      12 FORMAT(I1,I4,3I2,I4,2F9.6,F7.2,I3,F6.2,I5,2F6.1,F4.1,
245.      1 2I2,F6.1,I1,I2,35A1,I2)
246.      GO TO 900
247.      700 KK=-2
248.      KDEC=0
249.      CALL NAVBT(PLAT,PLONG,LAT,RLATM,KNS,
250.      1 LONG,RLONG,KEW,KDEC)
251.      LATM=RLATM
252.      LONG=RLONG
253.      CALL PINBT(ITAPE,JTAPE,KK,ISTA,KEY,LAT,LATM,KNS,
254.      1 LONG,LONG,KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,
255.      2 DINE,STHK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTW)
256.      GO TO 900
257.      800 KK=-2
258.      KL=-1
259.      CALL DNAV(DLAT,KSND,DLONG,KWE,PLAT,PLONG,KL)
260.      CALL YINBT(ITAPE,JTAPE,KK,
261.      1 ISR1,ISR2,KDA,KMB,KYR,KHM,SEC,DLAT,KSND,DLONG,KWE,DEPT,AMAG,IMB,
262.      2 ISOS,INTS,IDIAS,ITSU,ISEICH,IYBLC,INBNT,ING,IFEG,IMS,IASP,IZH,
263.      3 ICE,IMG,IAUTH,IGHY,NPP,IBQ,ILM,IS1,IS2)
264.      GO TO 900

```

```

265.      830  GO TO 1000
266.      900  IF(IENYR.EQ.0) GO TO 951
267.          CALL FIND(IENDA,IENM0,IENYR,IENHM,JDA,JM0,JYR,JHM,INDK)
268.          IF(INDK.EQ.0) GO TO 100
269.          GO TO 150
270.      951  GO TO 150
271.      992  GO TO 1000
272.      994  WRITE(IIBUT,993) JJREC
273.  993  FORMAT('END OF PROCESSING, RECORDS OUTPUT = ',I8)
274.          CALL EXIT
275.      995  WRITE(IIBUT,996) JDA,JM0,JYR,JHM
276.      996  FORMAT('END DATE PASSED',2X,3I2,I4)
277.          CALL EXIT
278.      999  WRITE(IIBUT,998) IND
279.      998  FORMAT('ERROR IN SKPREC,IND=',I,I3)
280.          CALL EXIT
281.      C
282.      C      END OF AN INPUT STREAM
283.      C
284.      1000 CONTINUE
285.      C
286.      C      WRITING AN 8 ON OUTPUT STREAM FOR INPUT TO MODPLOT PROGRAM
287.      C
288.      C
289.      C      IF JFMT = SEISMICITY OR VOLCANOES WRITE ONLY ON 8 , OTHERWISE
290.      C      WRITE TWO 8 'S.
291.      C
292.          IF (JFMT=4)1002,1008,1002
293.      1002 IF(JFMT=7)1007,1008,1007
294.      1007 WRITE(JTAPE,1001)I8
295.      1008 WRITE(JTAPE,1001)I8
296.          JFCT=JFCT+1
297.          IF(JF(JFCT))994,994,1005
298.      1005 JFMT=JF(JFCT)
299.      C
300.      C      WRITING JFMT NUMBER ON OUTPUT FOR INPUT TO MODPLOT PROGRAM
301.      C
302.          WRITE(JTAPE,1001)JFMT
303.      1001 FORMAT(I1,20X)
304.          GO TO 18
305.          END

```

NAME	TYPE	CLASS	HEX LBC	DEC WRDS	NAME	TYPE	CLASS	HEX LBC	DEC WRDS	NAME	TYPE	CLASS	HEX LBC	DEC WRDS	NAME	TYPE	CLASS	HEX LBC	DEC WRDS
AMAG	R	SCALR	000D6 V	1	AVMTN	R	SCALR	000C6 V	1	AVMTN	R	SCALR	000C6 V	1	AVMTN	R	SCALR	000C6 V	1
BG	R	SCALR	000C9 V	1	CRVN	R	SCALR	000C4 V	1	CRVN	R	SCALR	000C4 V	1	CRVN	R	SCALR	000C4 V	1
OBSC	R	SCALR	000F7 V	1	DEGRA	R	SCALR	00084 V	1	DEPT	R	SCALR	000D5 V	1	DEPT	R	SCALR	000D5 V	1
CINE	R	SCALR	000C2 V	1	DISR	R	SCALR	00088 V	1	DIST	R	SCALR	0008C V	1	DIST	R	SCALR	0008C V	1
DLAT	R	SCALR	000D1 V	1	DLBN	R	SCALR	000D3 V	1	DLNG	R	SCALR	00100 V	1	DLNG	R	SCALR	00100 V	1
DNAV	R	SPRGG	EXTERN	1	ELEV	R	SCALR	00097 V	1	EXTN	R	SPRGG	EXTERN	1	EXTN	R	SPRGG	EXTERN	1
FA	R	SCALR	00098 V	1	FIND	R	SPRGG	EXTERN	35	FLBAT	R	SPRGG	EXTERN	1	FLBAT	R	SPRGG	EXTERN	1
GINST	I	SPRGG	EXTERN	1	IA	I	ARRAY	00001 V	1	IASP	I	SCALR	000E2 V	1	IASP	I	SCALR	000E2 V	1
IAUTH	I	SCALR	000E6 V	1	ICE	I	SCALR	000E4 V	1	IDEP	I	SCALR	0009A V	1	IDEP	I	SCALR	0009A V	1
IDESC	I	ARRAY	000E2 V	6	IDIAS	I	SCALR	000DA V	1	IDIF	I	SCALR	00095 V	1	IDIF	I	SCALR	00095 V	1
IELC	I	SCALR	0009E V	1	IENDA	I	SCALR	000F2 V	1	IENHM	I	SCALR	000F5 V	1	IENHM	I	SCALR	000F5 V	1
IENMB	I	SCALR	000F3 V	1	IENYR	I	SCALR	000F4 V	1	IFBC	I	SCALR	000A3 V	1	IFBC	I	SCALR	000A3 V	1
IFEG	I	SCALR	000E0 V	1	IFFC	I	SCALR	000A2 V	1	IFLAG	I	SCALR	000E3 V	1	IFLAG	I	SCALR	000E3 V	1
IGC	I	SCALR	0009F V	1	IIN	I	SCALR	0007D V	1	IIBUT	I	SCALR	0007E V	1	IIBUT	I	SCALR	0007E V	1
ILM	I	SCALR	000EA V	1	IMANT	I	SCALR	000BA V	1	IMB	I	SCALR	000D7 V	1	IMB	I	SCALR	000D7 V	1
IMG	I	SCALR	000E5 V	1	IMS	I	SCALR	000E1 V	1	IND	I	SCALR	0008F V	1	IND	I	SCALR	0008F V	1
INDK	I	SCALR	000FF V	1	INIT	I	SCALR	0008E V	1	INNT	I	SCALR	000DE V	1	INNT	I	SCALR	000DE V	1
INTS	I	SCALR	000D9 V	1	I0GR	I	SCALR	000A4 V	1	I0G	I	SCALR	000E9 V	1	I0G	I	SCALR	000E9 V	1
IGHY	I	SCALR	000E7 V	1	I0RC1	I	SCALR	00083 V	1	I0GC	I	SCALR	000A1 V	1	I0GC	I	SCALR	000A1 V	1
ISEICH	I	SCALR	000DC V	1	ISKP	I	SCALR	000F6 V	1	ISL	I	SPRGG	INTRIN	1	ISL	I	SPRGG	INTRIN	1
ISBRC	I	SCALR	00096 V	1	IS0S	I	SCALR	000D8 V	1	ISR1	I	SCALR	000CA V	1	ISR1	I	SCALR	000CA V	1
ISR2	I	SCALR	000CB V	1	ISTA	I	SCALR	000B2 V	1	ISTDA	I	SCALR	000EE V	1	ISTDA	I	SCALR	000EE V	1
ISTHM	I	SCALR	000F1 V	1	ISTM9	I	SCALR	000EF V	1	ISTYR	I	SCALR	000FO V	1	ISTYR	I	SCALR	000FO V	1
ISW	I	SPRGG	EXTERN	1	ISI	I	SCALR	000EB V	1	IS2	I	SCALR	000EC V	1	IS2	I	SCALR	000EC V	1
ITAPE	I	SCALR	00090 V	1	ITSU	I	SCALR	000DB V	1	Iv6LC	I	SCALR	000DD V	1	Iv6LC	I	SCALR	000DD V	1
IM	I	ARRAY	00059 V	35	IWG	I	SCALR	000DF V	1	IYR	I	SCALR	000C1 V	1	IYR	I	SCALR	000C1 V	1
IZ	I	ARRAY	00050 V	9	IZH	I	SCALR	000E3 V	1	I8	I	SCALR	00082 V	1	I8	I	SCALR	00082 V	1
J	I	SCALR	00101 V	1	JDA	I	SCALR	000FB V	1	JF	I	ARRAY	00024 V	10	JF	I	ARRAY	00024 V	10
JFCT	I	SCALR	00080 V	1	JFMT	I	SCALR	000FC V	1	JHM	I	SCALR	000FE V	1	JHM	I	SCALR	000FE V	1
JJREC	I	SCALR	00081 V	1	JM0	I	SCALR	0009C V	1	JTAPE	I	SCALR	0007F V	1	JTAPE	I	SCALR	0007F V	1
JYR	I	SCALR	000FD V	1	KBG	I	SCALR	000A9 V	1	KCDM	I	SCALR	000AC V	1	KCDM	I	SCALR	000AC V	1
KCVC	I	SCALR	000AR V	1	KCVN	I	SCALR	000AA V	1	KDA	I	SCALR	000CC V	1	KDA	I	SCALR	000CC V	1
KDEC	I	SCALR	00102 V	1	KETV0	I	SCALR	000B1 V	1	KEM	I	SCALR	000B9 V	1	KEM	I	SCALR	000B9 V	1
KEY	I	SCALR	000B3 V	1	KFA	I	SCALR	000A8 V	1	KGDA	I	SCALR	00091 V	1	KGDA	I	SCALR	00091 V	1
KGHM	I	SCALR	00094 V	1	KGM0	I	SCALR	00092 V	1	KGYR	I	SCALR	00093 V	1	KGYR	I	SCALR	00093 V	1
KHM	I	SCALR	000CF V	1	KK	I	SCALR	00086 V	1	KL	I	SCALR	000FA V	1	KL	I	SCALR	000FA V	1
KMB	I	SCALR	000CD V	1	KNS	I	SCALR	000B6 V	1	KSN	I	SCALR	000D2 V	1	KSN	I	SCALR	000D2 V	1
KTAPE	I	SCALR	00000 V	1	KVE	I	SCALR	000A7 V	1	KVN	I	SCALR	000A6 V	1	KVN	I	SCALR	000A6 V	1
KWE	I	SCALR	000D4 V	1	KYR	I	SCALR	000CE V	1	K977	I	SCALR	00098 V	1	K977	I	SCALR	00098 V	1
LAT	I	SCALR	000B4 V	1	LATY	I	SCALR	000B5 V	1	L0M	I	SCALR	000B8 V	1	L0M	I	SCALR	000B8 V	1
LANG	I	SCALR	000B7 V	1	MAG1	I	SCALR	000AF V	1	MAG2	I	SCALR	000B0 V	1	MAG2	I	SCALR	000B0 V	1
MET	I	SCALR	000C0 V	1	MT	I	SCALR	000AE V	1	MTDC	I	SCALR	000AD V	1	MTDC	I	SCALR	000AD V	1
NAVIN	I	SPRGG	EXTERN	1	NAV9T	I	SPRGG	EXTERN	1	NELEV	I	SCALR	000BB V	1	NELEV	I	SCALR	000BB V	1
N0W	I	ARRAY	0004C V	4	NPP	I	SCALR	000E8 V	1	NW0N	I	SCALR	0007C V	1	NW0N	I	SCALR	0007C V	1
N1	I	SCALR	000BC V	1	N2	I	SCALR	000BD V	1	N3	I	SCALR	000BE V	1	N3	I	SCALR	000BE V	1
N4	I	SCALR	000CF V	1	N5	I	SCALR	000A5 V	1	NBSG	R	SCALR	00099 V	1	NBSG	R	SCALR	00099 V	1
PIN0T	I	SPRGG	EXTERN	1	PLAT	R	SCALR	00089 V	1	PL0NG	R	SCALR	0008A V	1	PL0NG	R	SCALR	0008A V	1
PR0J	I	SPRGG	EXTERN	1	RADEG	R	SCALR	00085 V	1	RFA	R	SCALR	000A0 V	1	RFA	R	SCALR	000A0 V	1
RLAT	R	SCALR	00087 V	1	RLATM	R	SCALR	000F8 V	1	RL0M	R	SCALR	000F9 V	1	RL0M	R	SCALR	000F9 V	1

RLNGT	R	SCALR	00088 V	1	SEC	R	SCALR	000D0 V	1	SETSKP	SPRGG	EXTERNAL	8
SINBT	R	SPRGG	EXTERNAL		SKPREC	R	SPRGG	EXTERNAL		STAT	SPRGG	EXTERNAL	
STHIK	R	SCALR	000C3 V	1	TC	R	SCALR	000D9 V	1	THICK	R	ARRAY	0003C V
TDAY	R	SPRGG	EXTERNAL		UNPKBY	R	SPRGG	EXTERNAL		VEL	R	ARRAY	00034 V
WGTN	R	SCALR	000C5 V	1	WGTN	R	SCALR	000C8 V	1	X	R	ARRAY	00044 V
YINBT	R	SPRGG	EXTERNAL										

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
2	0013B	8	00171	12	002E4	13	00022	16	00029
20	0004B	25	0005A	40	0006E	50	0008C	55	00093
82	0025C	90	00100	100	0012C	110	0012B	150	00174
190	00242	200	00180	210	001A0	300	001A1	310	001C0
410	001F4	500	00205	510	00232	530	0023E	599	0027A
608	00296	609	002A5	612	002C7	700	002F8	800	0032E
851	0026C	852	0026C	900	00365	951	00375	992	00376
994	00377	995	0038A	996	00392	998	003A2	999	0039D
1001	003C8	1002	003AF	1005	003C0	1007	003B2	1008	003B7

LABEL	HEX LBC	LABEL	HEX LBC
18	0004E	186	000DD
80	0003F	400	001CF
600	00287	600	00287
830	00364	993	0037C
1000	003AC	6365	0014F

## LOCAL VARIABLES (259 WORDS):

00000	KTAPE	00001	IA	00024	JF	0002E	IDESC	00034	VEL	0003C	THICK
00044	X	0004C	N84	0005C	I2	00059	IM	0007C	NWBN	0007D	IIN
00084	IIBUT	0007F	JTAPE	00080	JFCT	00081	JJREC	00082	I8	00083	IREC1
00084	DEGRA	00085	RADEG	00086	KK	00087	RLAT	00088	RLNG	00089	PLAT
0008A	PLNG	0008B	CISR	0008C	DIST	0008D	JFMT	0008E	INIT	0008F	IND
00090	ITAPE	00091	KGDA	00092	KGM0	00093	KGYR	00094	KGM	00095	IDIF
00096	ISARC	00097	ELEV	00098	K977	00099	BSG	0009A	IDEP	0009B	FA
0009C	BG	0009D	TC	0009E	IELC	0009F	IGC	000A0	RFA	000A1	IREGC
000A2	IFFC	000A3	IFRC	000A4	I8GR	000A5	B8	000A6	KVN	000A7	KVE
000A8	KFA	000A9	KBG	000AA	KCVN	000AB	KCVE	000AC	KCDM	000AD	MTDC
000AE	MT	000AF	MAG1	000B0	MAG2	000B1	KETV0	000B2	ISTA	000B3	KEY
000B4	LAT	000B5	LATM	000B6	KNS	000B7	LNG	000B8	LBM	000B9	KEW
000BA	IMANT	000BB	NELEV	000BC	N1	000BD	N2	000BE	N3	000BF	N4
000C0	MET	000C1	IYR	000C2	DINE	000C3	STHIK	000C4	CRVN	000C5	WGTN
000C6	AVMTN	000C7	CRVW	000C8	WGTW	000C9	AVMTW	000CA	ISR1	000CB	ISR2
000CC	KDA	000CD	KMB	000CE	KYR	000CF	KHM	000D0	SEC	000D1	DLAT
000D2	KSN	000D3	DLPN	000D4	KWE	000D5	DEPT	000D6	AMAG	000D7	IM3
000D8	ISBS	000D9	INTS	000DA	IDIAS	000DB	ITSU	000DC	ISEICH	000DD	IVBLC
000DE	INGT	000DF	ING	000E0	IFEG	000E1	IMS	000E2	IASP	000E3	I2H
000E4	ICE	000E5	IMG	000E6	IAUTH	000E7	IGHY	000E8	NPP	000E9	I83
000EA	ILM	000EB	IS1	000EC	IS2	000ED	IFLAG	000EE	IGTDA	000EF	ISTAB
000FC	ISTYR	000FD	ISTHM	000FE	IENDA	000FF	IENM0	000FA	IENYR	000FB	IENM1
000F6	ISKP	000F7	CRSG	000F8	RLATM	000F9	RLBM	000FA	KL	000FB	20A
000FC	JM9	000FD	JYR	000FE	JHM	000FF	INDK	00100	CLANG	00101	J
00102	KDEC										

BLANK COMPAN (0 WORDS)

## INTRINSIC SUBPROGRAMS USED:

FLBAT     ISL

## EXTERNAL SUBPROGRAMS REQUIRED:

DNAV	EXIT	FIND	GINBT	ISW	NAVIN	NAVBT	PINBT
PREJ	SETSKP	SINBT	SKPREC	STAT	TBDAY	UNPKBY	YINBT
F:101	F:102	F:103	F:104	F:105	F:106	F:108	M:DB
M:BC	9BCDREAD	9BCDWRT	9ENCDE	9ENDIOL	9INITIAL	9IOLDATA	9IOLUSA
9ITBR	9RTBI						

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS	
GENERATED CODE:	972	003CC	(NO MEMORY PROTECTION)
CONSTANTS:	7	00007	
LOCAL VARIABLES:	259	00103	
TEMPS:	0	00000	
TOTAL PROGRAM:	1238	004D6	

```

1. C PROGRAM SAINT2
2. C VERSION OF 7 AUG 1975, TO PROVIDE FOR MAGNETIC TAPE TAPE INPUT
3. C AND OUTPUT OF GSUM RECORDS
4. C VERSION OF 7 AUG 1974, TO IDENTIFY XSCAL
5. C VERSION OF 4 FEB 1973, TO REALLY CHANGE GSUM OUTPUT TO DEC DEG
6. C VERSION OF 8 SEPT 1972, TO CHANGE PCS DATA INPUT FORMAT
7. C AND TO ALLOW GSUM INPUT ON DEVICES OTHER THAN CARDS
8. C VERSION OF 21 AUGUST 1972, TO INPUT AND OUTPUT GSUM IN DEC DEG
9. C MOD MAY 20, 1971 BY FOLINSBEE TO FIT UNDER NEW MONITOR ARRAYS INTO COMMON.
10. C PROGRAM SAINT2, MODIFIED FROM SAINT ON 27 NOV 1970 BY C. BOWIN
11. C THIS IS A PROGRAM WRITTEN TO INTERPOLATE DATA AT EVEN INTERVALS
12. C DIMENSION A(10)
13. C DIMENSION NAME(5),SCALE(5),AMINV(5),AMAXV(5),Y(5)
14. C COMMON RLAT(1000),RLONG(1000),GRV(1000),FAIR(1000),BOUG(1000),
15. C 1 DEPTH(1000),ELEV(1000),DIST(1000),FATP(1000),HINT(1000)
16. C DIMENSION KEY(60)
17. C DIMENSION IAL(10),VAL(10),ARG(10)
18. C DIMENSION JA(10),JB(30)
19. C DOUBLE PRECISION G
20. C DATA NAME/'GRV ','FAIR','BOUG','DEPH','ELEV'/
21. C DATA KEY/'0008','0001','0001',' ',' ',' ','0','6',' ','01A0'
22. C A,'0730','08A','45',' ',' ',' ',' ',' ',' '
23. C DATA EPS/0.5/
24. C
25. C SSW(1) = 1 TO SORT DATA
26. C SSW(2) = 1 TO PUNCH SORTED DATA
27. C SSW(3) = 1 TO PRINT SORTED DATA
28. C SSW(4) = 1 TO INTERPOLATE VALUES
29. C SSW(5) = 1 TO PUNCH INTERPOLATED DATA IN GSUM FMT
30. C SSW(6) = 1 TO PUNCH INTERPOLATED FREE-AIR ANOMALY DATA IN TALPLOT FMT
31. C SSW(7) = 1 FOR PRINTED INTERPOLATED DATA AND SUMMARY OF JOB
32. C SSW(8) = 1 FOR PRINTING HEADING AND VALUES OF INPUT DATA
33. C SSW(9) = 1 TO PUNCH INTERPOLATED ELEVATION DATA IN TALPLOT FMT
34. C SSW(10) = 1 TO READ GSUM DATA FROM MAGNETIC TAPE
35. C SSW(11) = 1 TO WRITE INTERPOLATED DATA IN GSUM FMT ON MAGNETIC TAPE
36. C
37. C *****
38. C
39. C LAST INPUT GSUM DATA CARD SHOULD HAVE A 9 IN COLUMN 1
40. C
41. C *****
42. C
43. C SETUP INPUT AND OUTPUT DEVICES
44. C IIN=105
45. C IN=105
46. C IIBUT=108
47. C IBUT=108
48. C IPUNCH = 106
49. C JTAPE=2
50. C ITAPE=1
51. C IFILE1=3
52. C IFILE2=4
53. C IFILE3=5
54. C DD=0.0
55. C OUTPUT PROGRAM SAINT2, VERSION OF 7 AUG 1975
56. C DEGRA=1.745329E-02
57. C RADEG=57.29578
58. C INIT=ISW(=2)
59. C ENTER OPTIONS FOR PROCESSING

```

```

60.      5      READ(IIN,500,END=99) XKM,XSCAL,NPTS
61.      500    FORMAT(F5.1,F5.1,I5)
62.
63.      C      XKM = KM DISTANCE BETWEEN INTERPOLATED DATA POINTS
64.      C      XSCAL = MAX DISTANCE FOR WHICH POINTS ARE GIVEN UNITY WEIGHT-
65.      C      BEYOND THIS DROPS LIKE X/XSCAL.
66.      C
67.      IF(NPTS.EQ.0) NPTS=4
68.      NDIM=2*NPTS
69.      WRITE(IOUT,650) KEY
70.      650    FORMAT(T50,'INPUT PARAMETERS'//
71.      A      '  SORTING KEY USED',3(20A4//))
72.      OUTPUT XKM,XSCAL,NPTS
73.
74.      C      INITIALIZING FUNCTION WT
75.      C
76.      I=WTSET(XSCAL)
77.      C      READING PCS PARAMETERS AS INPUT TO SUBROUTINE PROJ
78.      READ(IIN,200)JA,ANG,DMAXM,ILAT,RILTM,ILONG,RILOM,JB
79.      200    FORMAT(10A1,2F10.0,I4,F6.2,I4,F6.2,30A1)
80.      WRITE(IOUT,200)JA,ANG,DMAXM,ILAT,RILTM,ILONG,RILOM,JB
81.      RILT=DMT0R(ILAT,RILTM)
82.      RILG=DMT0R(ILONG,RILOM)
83.      IF(ISW(8).EQ.1) WRITE(IOUT,604)
84.      604    FORMAT('O',T50,' INPUT DATA'//
85.      A1 RECORD      LATITUDE      LONGITUDE      GRAVITY      FREE AIR      BOUGUER      DE
86.      BPTH      ELEVATION TCOR      IELC// NUMBER      RADIANS      RADIANS      MG
87.      BALS      MGALS      MGALS      MTS      MTS      ')
88.      DO 6 INB=1,1000
89.      I=INB
90.      IF (ISW(10).EQ.0) GO TO 501
91.      READ (ITAPE,520,END=10)
92.      1 IREC,CLAT      ,DLONG      ,ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC
93.      520    FORMAT(I1,14X,2F9.4,F7.2,F9.2,F5.0,2F6.1,F4.1,I2)
94.      GO TO 505
95.      501    READ(ITAPE,502)
96.      1 IREC,CLAT      ,DLONG      ,ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC
97.      502    FORMAT(I1,14X,2F9.4,F7.2,F9.2,F5.0,2F6.1,F4.1,I2/)
98.      505    NB=INB-1
99.      RLAT(I)=DLAT*DEGRA
100.      RLONG(I)=DLONG*DEGRA
101.      GRV(INB)=G-977000.
102.      IF (IREC.EQ.9) GO TO 10
103.      IF (ISW(8).EQ.1) WRITE(IOUT,605)
104.      A 1,RLAT(I),RLONG(I),G,FAIR(I),BOUG(I),DEPTH(I),ELEV(I), TC,IELC
105.      605    FORMAT('I',1X, 15 ,4X,F9.6,2X,F9.6,1X,F9.2,2X,F6.1,4X,F6.1,3X,F6
106.      A.0,5X,F6.0,2X,F5.2,3X,I2)
107.      6      CONTINUE
108.      10     CONTINUE
109.      IREC=1
110.      NVAR=NB
111.      BSLAT=RLAT(1)
112.      BSLON=RLONG(1)
113.      T1=AMW(RLAT,NVAR)
114.      T2=AMW(RLAT,NVAR)
115.      T3=AMW(RLONG,NVAR)
116.      T4=AMW(RLONG,NVAR)
117.      DO 33 I=1,NVAR
118.      IF(RLAT(I).LT.BSLAT) BSLAT=RLAT(I),BSLON=RLONG(I)
119.      33     CONTINUE

```

```

120.      IF(T2-T1.GT.T4-T3) GO TO 34
121.      DO 37 I=1,NVAR
122.      IF(RLONG(I).LT.BSLON) BSLON=RLONG(I),BSLAT=RLAT(I)
123. 37      CONTINUE
124. 34      CONTINUE
125.      DO 31 I=1,N8
126.      CALL DISAZ(RLAT(I),RLONG(I),RILT,RILG,1,AZ,B,DISTKM,C)
127.      IF(ANG)44,44,42
128. 42      IF(AZ-135)48,48,46
129. 44      IF(AZ-225)48,48,46
130. 46      DISTKM=-DISTKM
131. 48      CONTINUE
132.      DIST(I)=DISTKM *10000.0
133. 31      CONTINUE
134.  C      ISW(1)=1 OR GREATER SORTS  0 OTHERWISE
135.      IF(ISW(1).EQ.0) GO TO 3C
136.      DO 11 I=1,NVAR
137.      G=GRV(I)+977000.
138.      WRITE(IFILE1,503)
139. 1 IREC,RLAT(I),RLONG(I),ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC,
140. 2 DIST(I)
141. 503  FORMAT(I1,14X,2F9.6,F7.2,F9.2,F5.0,2F6.1,F4.1,I2,F8.2)
142. 11      CONTINUE
143.  C      SORTING PART
144.      END FILE IFILE1
145.      REWIND IFILE1
146.      OUTPUT ' ', ' --SORTING TAKING PLACE-- '
147.      CALL CLOFIL(IFILE1)
148.  C
149.  C      IN THORT ARGUMENT LIST
150.      C      2ND ARG,  4  IS UNIT NO. FOR INPUT
151.      C      3RD ARG,  5  IS UNIT NO. FOR OUTPUT
152.  C
153.      CALL THORT(KEY,4,5,ICODE)
154.      OUTPUT ICODE
155.      CALL OPIN(IFILE3)
156.      OUTPUT ' --END OF SORT-- '
157.      DO 13 I=1,N8
158.      READ(IFILE3,503)
159. 1 IREC,RLAT(I),RLONG(I),ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC,
160. 2 DIST(I)
161.      GRV(I)=G+977000.
162. 13      CONTINUE
163.  C      CHOSE TO PRINT AND PUNCH DATA
164.  C      ISW(2)=1 TO PUNCH DATA
165.      IF(ISW(2).EQ.0) GO TO 3C
166.      DO 14 I=1,N8
167.      G=GRV(I)+977000.
168.      DLAT=RLAT(I)*RADEG
169.      DLONG=RLONG(I)*RADEG
170.      WRITE(IPUNCH,502)
171. 1 IREC,DLAT ,DLONG ,ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC
172. 14      CONTINUE
173. 30      CONTINUE
174.  C      IF (ISW(3)) =1 WANT PRINTED OUTPUT OF SORT
175.      IF(ISW(3).EQ.0) GO TO 35
176.      WRITE(18UT,602)
177. 602  FORMAT('1/' T40,'SORTED VALUES'/
178.  A' RECORDS  LATITUDE LONGITUDE GRAVITY FREE AIR BOUGUER DE
179.  BPTH  ELEVATION DISTANCE KM/' NUMBER  RADIANS  RADIANS  MG

```

```

180.      BALS      MGALS      MGALS      MTS      MTS      FROM #1')
181.      DO 32 I=1,N8
182.      G=GRV(I)+977000.
183.      ADIST=DIST(I)-10000.0
184.      WRITE(IOUT,603)
185.      A I,RLAT(I),RLONG(I),G,FAIR(I),BOUG(I),DEPTH(I),ELEV(I),ADIST
186. 603   FORMAT(' ',1X, 15 ,4X,F9.6,2X,F9.6,1X,F9.2,2X,F6.1,4X,F6.1,3X,F6
187.      A.0,5X,F6.0,3X,F8.2)
188.      CONTINUE
189.      C IF ISW(4) EQ 1 WILL INTERPOLATE
190.      35 IF (ISW(4).EQ.0) GO TO 5C
191.      C CALCULATING HEIGHT AND PLACING IT IN ELEV FIELD
192.      DO 45C I=1,N8
193.      IDEF=DEPTH(I)
194.      IF (IDEF)444,442,444
195.      442 HEIGHT=ELEV(I)
196.      ELEV(I)=HEIGHT
197.      GO TO 450
198.      444 HEIGHT =-DEPTH(I)
199.      ELEV(I)=HEIGHT
200.      45C CONTINUE
201.      IF (ISW(7).EQ.1) WRITE(IOUT,610)
202. 610   FORMAT('1'/T40,'INTERPOLATED VALUES')
203.      A' RECORD', ' GRAVITY FREE AIR BOUGUER DE
204.      BPTH ELEVATION DISTANCE KM ' /
205.      S ' NUMBER', ' MG
206.      BALS      MGALS      MGALS      MTS      MTS      FROM #1      NBRD
207.      * LATITUDE LONGITUDE' )
208.      C DETERMINING NEAREST FIELD POINT TO FIRST MEMBER OF DATA ARRAY
209.      IZER8=(DIST(1)-10000.)/XKM
210.      X=10000. + IZER8*XKM
211.      C ISW(5)=1 OUTPUT IN GSUM PUNCH
212.      C ISW(6)=1 OUTPUT IN TALPLT PUNCH
213.      C ISW(7)=1 PRINTED OUTPUT, INTERPOLATED
214.      NUMIT=0
215.      DO 36 I=1,1000
216.      IF (X.GT.DIST(N8))GO TO 436
217.      614 CONTINUE
218.      CALL PICK(X,DIST,N8,NPTS,IN,IZ)
219.      C FOR USE IN ATSM AND PICK THE SORTED ARRAY DIST MUST BE STORED
220.      C IN ORDER FROM THE SMALLEST TO LARGEST (IE. DIST(I)
221.      C MUST BE < OR = DIST(I+1) )
222.      IF (ABS(DIST(IN+1)-DIST(IN)).LE.0.001) RRLON=RLONG(IN)
223.      1 RRLAT=RLAT(IN), GO TO 644
224.      RRLON=(RLONG(IN) *(DIST(IN+1)-X)+RLONG(IN+1)*(X-DIST(IN)))
225.      1 /(DIST(IN+1)-DIST(IN))
226.      RRLAT=(RLAT (IN) *(DIST(IN+1)-X)+RLAT (IN+1)*(X-DIST(IN)))
227.      1 /(DIST(IN+1)-DIST(IN))
228.      644 IN=105
229.      NBR=NDIM
230.      CALL ATSM(X,DIST(IZ),NBR,ARG,IAL,NDIM)
231.      C THIS SECTION IS CALCULATING THE SUM OF THE WEIGHTS OF THE 3 OR 4
232.      C CLOSEST VARIABLES AND THE SUM OF ALL THE WEIGHTS
233.      C
234.      S3=WT(ARG(1)-X)+WT(ARG(2)-X)+WT(ARG(3)-X)
235.      S4=S3+WT(ARG(4)-X)
236.      S4E=0
237.      DO 664 IK=5,NDIM
238.      S4E=S4E+WT(ARG(IK)-X)
239.      664 CONTINUE

```

```

240.      S3E=S4E+WT(ARG(4)*X)
241. C WE START OUT WITH NORD (ORDER OF POLYGON) =3 . THEN IF THE FOLLOWING
242. C CONDITIONS ARE NOT MET WE REDUCE THE ORDER OF THE POLYGON
243. C THESE CONDITIONS ARE ONLY ESTIMATES, AND SHOULD PROBABLY BE
244. C CHANGED AS FURTHER EXPERIENCE IS GAINED WITH THE P PROGRAM
245.      NORD=3
246.      IF(S4.GT.4*S4E) NORD=2
247.      IF(S3.GT.4*S3E) NORD=1
248.      IF((S4+S4E).LE.4) NORD=2
249.      IF((S4+S4E).LE.2) NORD=1
250.      CALL SETAL(GRV(IZ),IAL,VAL,NDIM)
251.      CALL EGN(X,ARG,VAL,NDIM,NORD,A)
252.      GG=A(1)
253.      CALL SETAL(FAIR(IZ),IAL,VAL,NDIM)
254.      CALL EGN(X,ARG,VAL,NDIM,NORD,A)
255.      FF=A(1)
256.      CALL SETAL(BOUG(IZ),IAL,VAL,NDIM)
257.      CALL EGN(X,ARG,VAL,NDIM,NORD,A)
258.      BB=A(1)
259.      CALL SETAL(ELEV(IZ),IAL,VAL,NDIM)
260.      CALL EGN(X,ARG,VAL,NDIM,NORD,A)
261.      EE=A(1)
262.      FATP(I)=FF
263. C CONVERTING ELEVATION TO KMS FOR OUTPUT AT TALPLOT ELEV INPUT
264.      HINT(I)=EE*0.001
265.      G=GG+977000.
266.      IF(ISW(5).EQ.0) GO TO 705
267. C OUTPUT INTERPOLATED VALUES AT GSUM FMT ON TWO CARDS
268. C CONVERTING TO DECIMAL DEGREES
269.      DRLAT=RRLAT*RADEG
270.      DRLON=RRLON*RADEG
271.      IF(ISW(11).EQ.0) GO TO 699
272.      WRITE(CTAPE,696) DRLAT,DRLON,EE,G,DD,FF,BB
273. 696      FORMAT('1',14X,2F9.4,F7.0,F9.2,F5.0,2F6.1)
274.      GO TO 705
275. 699      WRITE(IPUNCH,700) DRLAT,DRLON,EE,G,DD,FF,BB
276. 700      FORMAT('1',14X,2F9.4,F7.0,F9.2,F5.0,2F6.1 /
277. 1      15X,'INTERPOLATED GSUM FROM SAINT2')
278. 705      XD=X-10000.
279. 41      IF(ISW(7).EQ.1) WRITE(ROUT,611) I,G,FF,BB,DD,EE,XD,NORD,RRLAT,RRLON
280.      *N
281. 611      FORMAT('1',1X,15,5X, F9.1,2X,F6.1,4X,F6.1,3X,F6
282. A.0,5X,F6.0,4X,F8.2,7X,13,2F10.6)
283.      X=X+ XKM
284.      NLIMIT=NLIMIT+1
285. 36      CONTINUE
286. 436      CONTINUE
287.      IF(ISW(6).EQ.1) WRITE(IPUNCH,615) (FATP(I),I=1,NLIMIT)
288. 615      FORMAT(5F10.1)
289.      IF(ISW(9).EQ.1) WRITE(IPUNCH,620) (HINT(I),I=1,NLIMIT)
290. 620      FORMAT(5F10.3)
291. 50      CONTINUE
292. 156      CONTINUE
293.      GO TO 5
294. 99      CONTINUE
295.      STOP
296.      END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
A	R	ARRAY	0000	10	ABS	R	SPRGG	0019	5	ADIST	R	SCALR	0007	1	ADIST	R	SCALR	0007	1
AXAM	R	SPRGG	0001	1	AMAXV	R	ARRAY	0020	1	AMINV	R	ARRAY	0014	5	AMINV	R	ARRAY	0014	5
ATSM	R	SPRGG	0002	1	ANG	R	SCALR	0021	1	ARG	R	ARRAY	0003	10	ARG	R	ARRAY	0003	10
BB	R	SCALR	0003	1	AZ	R	SCALR	0022	1	B	R	SCALR	0004	1	B	R	SCALR	0004	1
BSLN	R	SCALR	0004	1	BBUG	R	ARRAY	0023	1	BSLAT	R	SCALR	0005	1	BSLAT	R	SCALR	0005	1
DD	R	SCALR	0005	1	C	R	SCALR	0024	1	CLGFL	R	SPRGG	0006	1	CLGFL	R	SPRGG	0006	1
DISAZ	R	SPRGG	0006	1	DEGRA	R	ARRAY	0025	1	CLGFL	R	SPRGG	0007	1	CLGFL	R	SPRGG	0007	1
CLAT	R	SCALR	0007	1	DIST	R	SCALR	0026	1	DEPTH	R	SCALR	0008	1	DEPTH	R	SCALR	0008	1
DMT0R	R	SPRGG	0008	1	DLONG	R	SCALR	0027	1	DISTKM	R	SCALR	0009	1	DISTKM	R	SCALR	0009	1
EE	R	SCALR	0009	1	DRLAT	R	SCALR	0028	1	DMLXN	R	SCALR	0010	1	DMLXN	R	SCALR	0010	1
EGN	R	SCALR	0010	1	ELEV	R	SCALR	0029	1	DRLEN	R	SCALR	0011	1	DRLEN	R	SCALR	0011	1
FF	R	SCALR	0011	1	FAIR	R	ARRAY	0030	1	EPS	R	SCALR	0012	1	EPS	R	SCALR	0012	1
GRV	R	SCALR	0012	1	G	R	SCALR	0031	1	FATP	R	SCALR	0013	1	FATP	R	SCALR	0013	1
I	R	SCALR	0013	1	HEIGT	R	SCALR	0032	1	GG	R	SCALR	0014	1	GG	R	SCALR	0014	1
IDEP	R	SCALR	0014	1	IAL	R	SCALR	0033	1	HINT	R	SCALR	0015	1	HINT	R	SCALR	0015	1
IFILE2	R	SCALR	0015	1	IELC	R	SCALR	0034	1	IC8DE	R	SCALR	0016	1	IC8DE	R	SCALR	0016	1
IFILE3	R	SCALR	0016	1	IFILE3	R	SCALR	0035	1	IFILE1	R	SCALR	0017	1	IFILE1	R	SCALR	0017	1
ILONG	R	SCALR	0017	1	IK	R	SCALR	0036	1	ILAT	R	SCALR	0018	1	ILAT	R	SCALR	0018	1
IN0	R	SCALR	0018	1	IN	R	SCALR	0037	1	INIT	R	SCALR	0019	1	INIT	R	SCALR	0019	1
IREC	R	SCALR	0019	1	INOUT	R	SCALR	0038	1	IPUNCH	R	SCALR	0020	1	IPUNCH	R	SCALR	0020	1
I2	R	SCALR	0020	1	ISH	R	SCALR	0039	1	ITAPE	R	SCALR	0021	1	ITAPE	R	SCALR	0021	1
VB	R	SCALR	0021	1	IZERO	R	SCALR	0040	1	JA	R	SCALR	0022	1	JA	R	SCALR	0022	1
NAME	R	SCALR	0022	1	JTAPE	R	SCALR	0041	1	KEY	R	SCALR	0023	1	KEY	R	SCALR	0023	1
NOR	R	SCALR	0023	1	NDIM	R	SCALR	0042	1	NB	R	SCALR	0024	1	NB	R	SCALR	0024	1
NUMIT	R	SCALR	0024	1	NORC	R	SCALR	0043	1	NP7S	R	SCALR	0025	1	NP7S	R	SCALR	0025	1
PICK	R	SCALR	0025	1	NVAR	R	SCALR	0044	1	SPIN	R	SCALR	0026	1	SPIN	R	SCALR	0026	1
RILBM	R	SCALR	0026	1	RADEG	R	SCALR	0045	1	RILG	R	SCALR	0027	1	RILG	R	SCALR	0027	1
RLAT	R	SCALR	0027	1	RILT	R	SCALR	0046	1	RILTH	R	SCALR	0028	1	RILTH	R	SCALR	0028	1
RLON	R	SCALR	0028	1	RLONG	R	SCALR	0047	1	RRLAY	R	SCALR	0029	1	RRLAY	R	SCALR	0029	1
S3	R	SCALR	0029	1	SCALE	R	SCALR	0048	1	SETAL	R	SCALR	0030	1	SETAL	R	SCALR	0030	1
S4E	R	SCALR	0030	1	SCALE	R	SCALR	0049	1	THORT	R	SCALR	0031	1	THORT	R	SCALR	0031	1
T1	R	SCALR	0031	1	SEE	R	SCALR	0050	1	T3	R	SCALR	0032	1	T3	R	SCALR	0032	1
T4	R	SCALR	0032	1	TC	R	SCALR	0051	1	WT	R	SCALR	0033	1	WT	R	SCALR	0033	1
WTSET	R	SCALR	0033	1	T2	R	SCALR	0052	1	XD	R	SCALR	0034	1	XD	R	SCALR	0034	1
XKM	R	SCALR	0034	1	VAL	R	SCALR	0053	1	Y	R	SCALR	0035	1	Y	R	SCALR	0035	1
					XSCAL	R	SCALR	0054	1										

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
5	R	SCALR	0000	1	0000	R	SCALR	0000	1	0000	R	SCALR	0000	1	0000	R	SCALR	0000	1
30	R	SCALR	0001	1	0001	R	SCALR	0001	1	0001	R	SCALR	0001	1	0001	R	SCALR	0001	1
48	R	SCALR	0002	1	0002	R	SCALR	0002	1	0002	R	SCALR	0002	1	0002	R	SCALR	0002	1
42	R	SCALR	0003	1	0003	R	SCALR	0003	1	0003	R	SCALR	0003	1	0003	R	SCALR	0003	1
503	R	SCALR	0004	1	0004	R	SCALR	0004	1	0004	R	SCALR	0004	1	0004	R	SCALR	0004	1
605	R	SCALR	0005	1	0005	R	SCALR	0005	1	0005	R	SCALR	0005	1	0005	R	SCALR	0005	1
644	R	SCALR	0006	1	0006	R	SCALR	0006	1	0006	R	SCALR	0006	1	0006	R	SCALR	0006	1
705	R	SCALR	0007	1	0007	R	SCALR	0007	1	0007	R	SCALR	0007	1	0007	R	SCALR	0007	1

LOCAL VARIABLES (238 WORDS):

0000 A	0000 NAME	0001E Y
0003 KEY	0005F IAL	00087 JB

```

000A6 G
000A7 IFLACH
000B3 CD
000B4 APTS
000B5 RILTM
000C5 IREC
000C6 NVAR
000C7 T4
000D7 ADIST
000CC IZ
000E3 S4E
000E9 BB

000A8 EPS
000A9 WTape
000B4 DEGRA
000B5 NDIM
000C0 ILONG
000C6 DLAT
000C7 BSLAT
000C8 AZ
000D2 AZ
000D8 IDEP
000DE RRL9N
000E4 IK
000EA EE

000A9 IIN
000AF ITAPE
000B5 RADEG
000B6 I
000C1 RIL8M
000C7 DLONG
000CC BSL8N
000C3 B
000D9 HEIGT
000CF RRLAT
000E5 S3E
000EB DRLAT

000AA IN
000B0 IFILE1
000B6 INIT
000B7 ANG
000C2 RILT
000C3 TC
000C8 T1
000D4 DISTKM
000DA IZER0
000EC N8R
000E6 N8RD
000EC DRL8N

000AB IIBUT
000B1 IFILE2
000B7 XKM
000B8 DMAXM
000C3 RILG
000C9 IELC
000CF T2
000D5 C
000DB X
000E1 S3
000E7 GG
000ED XD

000AC IOUT
000B2 IFILE3
000B8 XSCAL
000BE ILAT
000C4 IN8
000CA N8
000D0 T3
000D6 IC8DE
000DC NUMIT
000E2 S4
000E8 FF

```

## BLANK COMMON (1000 WORDS):

```

00000 RLAT      003E8 RLONG
0177C ELEV      01B58 DIST

007DC GRV
01F4C FATP

008B8 FAIR      00FA0 BUG      01388 DEPTH
02328 HINT

```

## INTRINSIC SUBPROGRAMS USED:

ABS

## EXTERNAL SUBPROGRAMS REQUIRED:

```

AMIN  AMIN  ATSM
SPIN  PICK  SETAL
F:103 F:104 F:105
9BCDREAD 9BCDWRIT 9DTER 9REWIND 9IT8R

CLOFIL TH8RT F:106 9ENDFILE 9RT8I 9ST8P
DISAZ WT F:108 9ENDI8L 9ST8P

DMT8R WTSET MID8 SINITIAL
EQU F:101 M:8C 9I8DATA
ISM F:102 9BCDRDEE 9I8LUSA

```

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

```

DEC WORDS 1308
HEX WORDS 0051C
GENERATED CODE: 1308
CONSTANTS: 24
LOCAL VARIABLES: 238
TEMPS: 5
TOTAL PROGRAM: 1575
(PLUS BLANK COMMON)

```

```

1.  C   PROGRAM SELSP
2.  C   VERSION OF 11 AUG 75 (ORIGINAL)
3.  C   OUTPLT 'PROGRAM SELSP' = VERSION OF 11 AUG 75'
4.  C   FOR SELECTING SPFMT OUTPLT OF CRWT3 (DERIVED FROM PROGRAM SORT3)
5.  C
6.  C   PROGRAM SORT3, FOR SORTING OUTPUT OF CRWT2
7.  C
8.  C   DIMENSION IDESC(6),VEL(8),THICK(8)
9.  C
10. C   INITIALIZATION
11. C
12. C   IIN = 105
13. C   IIOUT = 108
14. C   ITAPE = 1; JTAPE = 2
15. C   NREC = 0
16. C   IOREC = 0
17. C
18. C   SENSE SWITCH CARD IS REQUIRED BY PINOT
19. C
20. C   INIT = ISW(-2)
21. C   KK = 0
22. C   CALL PINOT (ITAPE,JTAPE,KK,ISTA,KEY,LAT,LATM,KNS,LONG,LON,
23. C   1KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,
24. C   2STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTH)
25. C   ISTAB = 0
26. C   K9 = 1H9
27. C   NNS = 1HS
28. C   NEW = 1HW
29. C
30. C
31. C   READ(IIN,2) JMET,IMET,JPROV,IPROV,IAR,IMDP,SMIND,SMAXD
32. C   2 FORMAT(6I5,2F10.0)
33. C   IDP = INDICATOR FOR SORT ON WATER DEPTH OR ELEVATION
34. C   DMIN = MINIMUM WATER DEPTH OR ELEVATION
35. C   DMAX = MAXIMUM WATER DEPTH, OR ELEVATION
36. C   READ(IIN,3) IDP,DMIN,DMAX
37. C   3 FORMAT (I5,2F10.0)
38. C   OUTPUT JMET,IMET,JPROV,IPROV,IAR,IMDP,SMIND,SMAXD,IDP,DMIN,DMAX
39. C   IF(IAR.GT.0) CALL ARLIM (IIN,IIOUT,RTOP,RBOT,RLEFT,RRIGT)
40. C
41. C   READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
42. C
43. C   10 CONTINUE
44. C   KK = 1
45. C   CALL PINOT (ITAPE,JTAPE,KK,ISTA,KEY,LAT,LATM,KNS,LONG,LON,
46. C   1KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,
47. C   2STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTH)
48. C   NREC = NREC + 1
49. C   END-OF-FILE OR END-OF-TAPE ENCOUNTERED DURING READ
50. C   IF(KK.EQ.9) GO TO 999
51. C   CHECKING IF KEY = 9
52. C   18 IF(KEY=K9) 20,10,20
53. C   20 CONTINUE
54. C   CHECKING FOR SORT CHECKS
55. C   ISTAB = ISTA
56. C   IF(JMET) 300,310,300
57. C   300 IF(IMET=MET) 10,310,10
58. C   310 IF(JPROV) 315,320,315
59. C   315 NPROV = N1*1000 + N2*100 + N3*10 + N4

```

```

60.      IF (IPR8V-NPR8V)10,320,10
61.      320  IF (IMDP)325,330,325
62.      325  IF (DINE-SMIND)10,327,327
63.      327  IF (SMAXD-DINE)10,330,330
64.      330  IF (IAR)335,360,335
65.      335  RLATM=LATM
66.      RL8M=L8M
67.      RLAT=DMT8R(LAT,RLATM)
68.      RL8G=DMT8R(L8G,RL8M)
69.      IF (KNS=NNS)354,352,354
70.      352  RLAT=RLAT
71.      354  IF (KEW=NEW)358,356,358
72.      356  RL8G=RL8G
73.      358  CALL ARCK(RLAT,RL8G,RT8P,R88T,RLEFT,RRIGT,IND)
74.      IF (IND)10,360,10
75.      360  IF (IDP) 364,370,362
76.      C    SEA SEISMIC PROFILE
77.      362  IF (N1.EG.2) ELEV=NELEV; GO TO 366
78.      GO TO 10
79.      C    LAND SEISMIC PROFILE
80.      364  IF (N1.EG.2) GO TO 10
81.      ELEV=NELEV
82.      366  IF (ELEV=DMIN)10,368,368
83.      368  IF (DMAX=ELEV)10,370,370
84.      370  CONTINUE
85.      C
86.      C    OUTPUT RESULTS
87.      KK=2
88.      CALL PIN8T (ITAPE,UTAPE,KK,ISTA,KEY,LAT,LATM,KNS,L8G,L8M,
89.      1KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,
90.      2STHIK,CRVN,WGTN,AVWTN,CRVH,WGTW,AVWTW)
91.      I8REC=I8REC+1
92.      GO TO 10
93.      999  CONTINUE
94.      WRITE(II8UT,600) NREC,I8REC
95.      600  FORMAT ('NUMBER 8F RECORDS INPUT=',I10,
96.      1 'NUMBER 8F RECORDS 8UTPUT=',I10)
97.      IF (ISW(26))40,35,40
98.      35  END FILE UTAPE
99.      40  CALL EXIT
100.     END

```

[illegible]

## LOCAL VARIABLES (80 WORDS):

[illegible]

BLANK COMMON (0 WORDS)

EXTERNAL SUBPROGRAMS REQUIRED!

ARCK ARLIM DMYGR  
F:1C3 F:1C4 F:105

EXIT 15W  
F:106 F:108

FINOT  
NIDB

F:101  
M:9C

F:102  
9BCDREAD

SBCWRIT SENDFILE SENDI8L 9INITIAL 9I8DATA 9I70R 9PRINT 9STOP

HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	370	00172
CONSTANTS:	5	00005
LOCAL VARIABLES:	80	00050
TEMPS:	2	00002
TOTAL PROGRAM:	457	001C9

COMPILED 20 MAR 73

```

1.      C      PROGRAM SPFMT
2.      C
3.      C      VERSION OF 21 AUGUST 1971, TO READ NEW CARD SET REC'D JULY 71
4.      C      DIMENSION IDESC(6),VEL(8),THICK(8),X(8)
5.      C      DIMENSION JDESC(6)
6.      C
7.      C      PROGRAM SPFMT, CONVERTS SEISMIC REFRACTION COLUMN FORMAT
8.      C      FROM U OF TORONTO WORLD SEISMIC REFRACTION COMPIATION
9.      C
10.     C      FORMAT TO SPFMT FORMAT
11.     C      EACH PROFILE IN ONE 120 CHARACTER RECORD
12.     C
13.     C      USES SUBROUTINES ISW, STAT, EVIL
14.     C
15.     C
16.     C      *****
17.     C      ITAPE = URN FOR SEISMIC DATA INPUT
18.     C      JTAPE = URN FOR SPFMT DATA OUTPUT
19.     C      ITAPE = 1
20.     C      JTAPE = 2
21.     C      *****
22.     C
23.     C
24.     C
25.     C      ***** NOTE *****
26.     C
27.     C      IN ORDER TO OUTPUT THE LAST DATA CARD, A BLANK CARD MUST
28.     C      FOLLOW THE DATA CARDS.
29.     C
30.     C      *****
31.     C
32.     C      IIN = 105
33.     C      IIOUT = 108
34.     C      CALL STAT
35.     C      INN = ISW(*2)
36.     C      WRITE (IIOUT,605)
37.     C      605  FORMAT ( / 'PROGRAM SPFMT' // )
38.     C      OUTPUT ' VERSION OF 21 AUGUST 1971'
39.     C      ISTA=0
40.     C      RND=0.5
41.     C      NREC=0
42.     C      IFST=0
43.     C
44.     C      L9 = 9
45.     C      L0 = 0
46.     C      L3 = 3
47.     C      L4 = 4
48.     C      KNUM = #
49.     C
50.     C      L9 = 1H9
51.     C      L0 = 1H0
52.     C      L3 = 1H3

```

```

53.      L4 = 1H4
54.      KNUM = 1H#
55.      L2 = 1H2
56.      L6 = 1H6
57.      L8 = 1H8
58.      LK = 1HK
59.      LM = 1HM
60.      LP = 1HP
61.      LR = 1HR
62.      LT = 1HT
63.      LV = 1HV
64.      LX = 1HX
65.      LZ = 1HZ
66.      LQ = 1H
67.      IREC = 1
68.      DINE = 0.0
69.      STWIK = 0.0
70.      CRVN = 0.0
71.      WGTN = 0.0
72.      AVWTN = 0.0
73.      CRVW = 0.0
74.      WGTW = 0.0
75.      AVWTW = 0.0
76.      C
77.      C
78.      KDA = 0
79.      KM0 = 0
80.      KYR = 0
81.      C READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
82.      999 CONTINUE
83.      READ (ITAPE, 12) JISTA, JKEY, JLAT, JLATM, JKNS, JLONG, JLOM,
84.      1 JKEW, V1, T1, V2, T2, V3, T3, V4, T4, VMATJ, ELEVJ,
85.      2 JN1, JN2, JN3, JN4, JMET, JIYR, JDESC
86.      12 FORMAT (I4, A1, I2, I2, A1, I3, I2, A1, 4 (F2.1, F3.1),
87.      1 F2.1, 17X, F4.2, 4 I1, I1, I2, 6A2)
88.      CALL STAT(I)
89.      CALL EVIL (IIOUT, I, IBAD, KDA, KM0, KYR, JISTA)
90.      IF (IBAD) 999, 13, 995
91.      13 CONTINUE
92.      C
93.      C CHECKING FOR KEY CODE FOR SECOND CARD
94.      C
95.      IF (JKEY = L2) 501, 550, 501
96.      501 IF (JKEY = L6) 502, 550, 502
97.      502 IF (JKEY = L8) 503, 550, 503
98.      503 IF (JKEY = LK) 504, 550, 504
99.      504 IF (JKEY = LM) 505, 550, 505
100.     505 IF (JKEY = LP) 506, 550, 506
101.     506 IF (JKEY = LR) 507, 550, 507
102.     507 IF (JKEY = LT) 508, 550, 508
103.     508 IF (JKEY = LV) 509, 550, 509
104.     509 IF (JKEY = LX) 510, 550, 510
105.     510 IF (JKEY = LX) 511, 550, 511

```

```

106.      511 IF(JKEY=LZ)512,550,512
107.      512 IF(JKEY=LQ)513,550,513
108.      513 GO TO 301
109.      550 IF(JISTA=ISTA0)444,555,444
110.      444 ISTA0=0
111.      GO TO 999
112.      555 CONTINUE
113.      C
114.      C CONVERT READ VALUES TO THOSE FOR LAYERS 5 = 8
115.      C
116.      VEL(5) = V1
117.      VEL(6) = V2
118.      VEL(7) = V3
119.      VEL(8) = V4
120.      THICK(5) = T1
121.      THICK(6) = T2
122.      THICK(7) = T3
123.      THICK(8) = T4
124.      C TAKING INFO FROM SECOND CARD FOR OUTPUT
125.      KEY=JKEY
126.      LAT=JLAT
127.      LATM=JLATM
128.      KNS=JKNS
129.      LONG=JLONG
130.      LOM=JLOM
131.      KEW=JKEW
132.      VMANT=VMATJ
133.      IMANT = VMANT*10.0
134.      ELEV=ELEVJ
135.      NELEV = ELEV * 100.0 +(SIGN(RND,ELEV))
136.      N1=JN1
137.      N2=JN2
138.      N3=JN3
139.      N4=JN4
140.      MET=JMET
141.      IYR=JIYR
142.      DO 560 I = 1,6
143.      IDESC(I)=JDESC(I)
144.      560 CONTINUE
145.      C SETTING INDICATOR FOR PROCESS CONTROL AFTER OUTPUT
146.      IND=2
147.      IFST=0
148.      600 CONTINUE
149.      85 J1 = VEL(1) *10.0 + 0.5
150.      J2 = VEL(2) *10.0 + 0.5
151.      J3 = VEL(3) *10.0 + 0.5
152.      J4 = VEL(4) *10.0 + 0.5
153.      J5 = VEL(5) *10.0 + 0.5
154.      J6 = VEL(6) *10.0 + 0.5
155.      J7 = VEL(7) *10.0 + 0.5
156.      J8 = VEL(8) *10.0 + 0.5
157.      K1 = THICK(1) *10.0 + 0.5
158.      K2 = THICK(2) *10.0 + 0.5

```

```

159.      K3 = THICK(3)      *10.0 + 0.5
160.      K4 = THICK(4)      *10.0 + 0.5
161.      K5 = THICK(5)      *10.0 + 0.5
162.      K6 = THICK(6)      *10.0 + 0.5
163.      K7 = THICK(7)      *10.0 + 0.5
164.      K8 = THICK(8)      *10.0 + 0.5
165.      WRITE(JTAPE,990) IREC1, ISTA, KEY, LAT, LATM, KNS, LONG, LOM,
166.      1 KEW, J1, K1, J2, K2, J3, K3, J4, K4, J5, K5, J6, K6, J7, K7, J8, K8,
167.      2 IMANT, NELEV, N1, N2, N3, N4, MET, IYR, IDESC, DINE, STHIK, CRVN
168.      3 WGTN, AVWTN, CRVW, WGTW, AVWTW
169.      990 FORMAT(I1, I4, A1, I2, I2, A1, I3, I2, A1, 8(I2, I3), I2, I4, 4I1,
170.      1 I1, I2, 6A2, 2F4.1, F3.1, 2F6.0, 1X, F3.1, 2F6.0, 5X)
171.      NREC=NREC+1
172.      C SETTING VELOCITY AND THICKNESS ARRAYS = ZERO
173.      DO 602 I = 1, 8
174.      VEL(I) = 0.0
175.      THICK(I) = 0.0
176.      602 CONTINUE
177.      610 GO TO (330, 610) IND
178.      610 ISTAR=0
179.      GO TO 999
180.      301 IF(JKEY=LG) 310, 305, 310
181.      305 IF(V1=0.01) 999, 999, 310
182.      310 IF(IFST) 312, 320, 312
183.      312 IND=1
184.      GO TO 600
185.      320 IFST=1
186.      330 CONTINUE
187.      C
188.      C CONVERT READ VALUES TO THOSE FOR LAYERS 1 = 4
189.      C
190.      ISTAR=JISTAR
191.      KEY=JKEY
192.      LAT=JLAT
193.      LATM=JLATM
194.      KNS=JKNS
195.      LONG=JLONG
196.      LOM=JLOM
197.      KEW=JKEW
198.      VMANT=VMATJ
199.      IMANT = VMANT*10.0
200.      ELEV=ELEVJ
201.      NELEV = ELEV * 100.0 + (SIGN(RND, ELEV))
202.      N1=JN1
203.      N2=JN2
204.      N3=JN3
205.      N4=JN4
206.      MET=JMET
207.      IYR=JIYR
208.      DO 340 I = 1, 6
209.      IDESC(I)=JDESC(I)
210.      340 CONTINUE
211.      22 VEL(I) = V1

```

```

212.          VEL(2) = V2
213.          VEL(3) = V3
214.          VEL(4) = V4
215.          THICK(1) = T1
216.          THICK(2) = T2
217.          THICK(3) = T3
218.          THICK(4) = T4
219.          ISTAO=JISTA
220.  C      READ NEW INPUT CARD
221.          GO TO 999
222.  995     END FILE JTAPE
223.          WRITE(IIOU,996)NREC
224.  996     FORMAT(1E0F FOUND, NREC = 1, I8)
225.          CALL EXIT
226.          END

```

LABEL	HEX L6C	LABEL	HEX L6C	LABEL	HEX L6C	LABEL	HEX L6C	LABEL	HEX L6C
12	0087	13	00A8	22	0C2C	85	001D	301	001C
310	00100	312	00102	320	0C1D5	330	001D7	340	002A
507	000AB	508	000A5	503	0C081	504	000B4	505	000B7
507	000B0	508	000C0	509	0C0C3	510	000C6	511	000C9
513	000CF	550	000D0	555	CC0D6	560	00117	600	0011D

AD-A035 454

WHOI-77-2  
GRAVITY DATA PROCESSING PROGRAMS.(U)

FEB 77

CARL BOWIN

UNCLASSIFIED WOODS HOLE OCEANOGRAPHIC INSTITUTION, MA

4 OF 6  
AD-A  
035 454



00000	IDESC	00006	VEL	0000E	THICK	00016	X	0001E	JDESC	00024	ITAPE
00005	JTAPE	00026	IN	00027	IUBT	00028	INN	00029	ISTAB	0002A	RND
00008	NREC	00032	IFST	00033	L9	00034	L8	00035	LK	00036	L4
00031	KNUM	00032	L2	00033	L6	00034	L8	00035	LK	00036	L4
00037	LP	00038	LR	00039	LT	0003A	LV	0003B	LX	0003C	L2
0003D	LQ	0003E	IREC1	0003F	DINE	00040	STHIK	00041	CRVN	00042	HGTN
00043	AVMTN	00044	CRVM	00045	HGTW	00046	AVMTW	00047	KDA	00048	KGN
00049	KYR	0004A	JISTA	0004B	JKEY	0004C	JLAT	0004D	JLATH	0004E	KJNS
0004F	JLNG	00050	JLGM	00051	JKEW	00052	V1	00053	T1	00054	V2
00055	T2	00056	V3	00057	T3	00058	V4	00059	T4	0005A	VMATJ
0005B	ELEVJ	0005C	YN1	0005D	YN2	0005E	JN3	0005F	JN4	00060	JMET
00061	J1YR	00062	IBAD	00063	IBAD	00064	KEY	00065	JAT	00066	JLATH
00067	KNS	00068	LANG	00069	LEM	0006A	KEY	0006B	VMANT	0006C	IMANT
0006D	ELEV	0006E	NELEV	0006F	N1	00070	N2	00071	N3	00072	N4
00073	MET	00074	IYR	00075	IND	00076	J1	00077	J2	00078	J3
00079	J4	0007A	J5	0007B	J6	0007C	J7	0007D	J8	0007E	K1
0007F	K2	00080	K3	00081	K4	00082	K5	00083	K6	00084	K7
00085	K8	00086	ISTA								

BLANK COMMON (0 WORDS)

### INTRINSIC SUBPROGRAMS USED:

## SIGN

**EXTERNAL SUBPROGRAMS REQUIRED:**

EVIL  
F:105  
9ENDIOL

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC	HEX
	WORDS	WORDS
GENERATED CODE:	562	00232
CONSTANTS:	22	00016
LOCAL VARIABLES:	135	00087
TEMPS:	1	00001
TOTAL PROGRAM:	720	002D0

COMPILED 24 April 1975

```

1.  C   PROGRAM TALPL0T 16
2.  C   OUTPUT 1 TALPL0T 16 RUN, VERSION 0F 8 APRIL 75
3.  C   VERSION 8 APRIL 75 TO ZERO VARIABLES
4.  C   VERSION 0F 25 MAY 1973, CHANGING TALPL0T 15 TO 16 AND
5.  C   CORRECTING ERROR IN 2D BOUGUER ANOMALY (BGT0) CALCULATION
6.  C   BGT0 VALUES WERE ONLY CORRECT IN PAST WHEN RDENS EQUALLED
7.  C   THE VALUE 2.67.
8.  C   VERSION 0F NOV15 TO CORRECT OUPUT 0N JTAPE 0F NEW FIELD POINTS
9.  C   SO THAT PLOTTING PROGRAM TERMINATES CORRECTLY
10. C   MOD PCT 14 TO CORRECT WEIGT BUG AND TO REMOVE PUNCHING 0F
11. C   MODIFIED PRINTS. TIMER FEATURE ADDED
12. C   VERSION 0F SEPT 28 CORRECTED WT CALCULATION FOR ELEVATION
13. C   AND IMPLEMENTED FILTERING 0F RESIDUALS
14. C   MODIFIED SEPT 14, 1971 BY FOLINSBEE
15. C   AND TO FILTER THE RESIDUAL BETWEEN OBSERVED AND THEORETICAL GRAVITY
16. C   MOD 0F JUNE 9, 1971, COMENST 0N POLYGON CARDS, CORRECT WT CALC.
17. C   MODIFIED JUNE 2 71 TO USE ELEVATIONS IN CRUSTAL WT CALCULATION
18. C   VERSION 0F 5 APRIL 1971, WRITES COMPLETE BOUGUER 0N JTAPE
19. C   LAST CHANGE FEB 3 71 TO READING 0F ELEVATIONS
20. C   C LAST CHANGE JAN 19 71 TO REMOVE 0GGA BUG
21. C   LAST CHANGE (REMOVE 0GA SET TO 0. BUG) 0N JAN 14, 71 AFOLINSBEE
22. C   TALPL0T 15 FROM TALPL0T 14 0N OCT 23, 1970 BY AFOLINSBEE
23. C   THIS IS A PROGRAM TO 0F 00TH LAND AND SEA DATA
24. C   ELEVATION MUST FOLLOW 0GA DAT, BEGINNING WITH A NEW CARD IN FORMAT
25. C   SF10.2
26. C   FOR USE 0N SIGMA 7, 7 TRACK MAGNETIC TAPE
27. C   OUTPUT REPRODUCES INPUT AND ALSO RESULTS
28. C
29. C   ASSIGN 9 TRACK MAG TAPE TO UNIT NUMBER 2
30. C   LAST BODY POINT IN EACH POLYGON MUST HAVE A 9 IN CBL 21
31. C   WEIGH EXPECTS THE DIMENSION 0F X,Z, TO 00 BE 3 .GT. THE #0F POLY
32. C   PRINTS
33. C
34. C   LNB=1 SHOULD BE FOR WATER LAYER ONLY
35. C   IF LNB=1, THEN 2-D BOUGUER ANOMALY IS CALCULATED.
36. C   REFERENCE DENSITY AND WEIGHT INPUTS REQUIRED.
37. C   THE WEIGHT IS CALCULATED FOR THE INPUT POLYGONS, Z AND N9T
38. C   FOR THE MODIFIED POLYGONS
39. C
40. C   ISW(1)=0 WRITE JTAPE
41. C   #1 NO WRITE JTAPE
42. C
43. C   ISW(2)=0 SETS ELEV =0
44. C   #1 READS IN ELEV IN KM
45. C   #2 READS IN ELEV IN METERS
46. C
47. C   ISW(3)=0 PRINTS INTERMEDIATE DATA FOR EACH POLYGON
48. C   #1 NO PRINTOUT 0F INTERMED DATA
49. C
50. C   ISW(4)=0 PLOT INTERMEDIATE DATA
51. C   #1 NO PLOT
52. C
53. C   ISW(5)=0 NO PLOT 0F ELEVATION
54. C   #1 PLOT ELEVATION
55. C   ISW(6) =1 TO ADJUST LAST CURVE TO FIT GRAVITY DATA
56. C   SSW(7) UP FOR OUTPUT DURING DEBUGGING ONLY
57. C
58. C   ISW(8) =1 TO NOT WRITE INTERMEDIATE DATA FOR EACH POLYGON 0NT9
59. C   JTAPE

```

```

TAU00000
TAU00010
TAU00040
TAU00050
TAU00060
TAU00070
TAU00080
TAU00090
TAU00100
TAU00110
TAU00120
TAU00130
TAU00140
TAU00150
TAU00160
TAU00170
TAU00180
TAU00190
TAU00200
TAU00210
TAU00220
TAU00230
TAU00250
TAU00260
TAU00270
TAU00280
TAU00290
TAU00300
TAU00310
TAU00320
TAU00330
TAU00340
TAU00370
TAU00380
TAU00390

```

```

60. C
61. C SSW(11) UP TO NOT USES ELEVATION VALUES IN THE CALCULATION
62. C OF THEORETICAL GRAVITY
63. C IF SSW(11) IS UP THEN THE GRAVITY VALUES READ IN SHOULD
64. C BE COMPLETE BOUGUER ANOMALIES
65. C IALTE = 1 MEANS THAT THIS POINT WILL BE VARIED TO COMPUTE A BEST FIT TAU000400
66. C SSW(12) UP TO USE ELEVATION VALUES IN CRUSTAL WT CALCULATION
67. C THIS SHOULD BE USED WHEN THE GRAVITY ANOMALIES ARE BOUGUER ANOMALIES
68. C AND THUS THE TOP OF THE MODEL IS AT SEA LEVEL. A DENSITY OF 2.67
69. C IS USED IN MAKING THE WT CORRECTION
70. C SSW(13) = 1 TO OUTPUT RESULTS OF INPUT POLYGONS BEFORE
71. C ALTERING THE VARIABLE BOUNDARY POINTS
72. C THE POLYGON THAT IS TO BE VARIED MUST BE THE LAST POLYGON TO BE TAU000410
73. C THE POINT TO BE VARIED MUST NOT BE THE FIRST OR LAST POINT IN THE TAU000420
74. C POLYGON TAU000430
75. C IMAX IS THE MAXIMUM NUMBER OF MODELS THAT WILL BE CALCULATED TAU000440
76. C IMOD IS THE NUMBERR OF MODELS THAT HAVE BEEN CALCULATED TAU000450
77. C TAU000460
78. C LOGICAL BAR/.FALSE./ TAU000470
79. C COMMON FER(200), WFER(-10:10)
80. C DIMENSION LABEL(20) TAU000480
81. C DIMENSION COME(5)
82. C DIMENSION FX(200), FZ(200) TAU000490
83. C DIMENSION PDELZ(200), SSELZ(200), XO(1), X(200), ZO(1), Z(200), TAU000500
84. C 1 BGA(200), RESA(200), TEST(200), DSU(200) TAU000510
85. C DIMENSION ARRAY (200,5), SUM(200), DWGT(200), PCEN(200), SUMW(200) TAU000520
86. C DIMENSION PFDEL(20), XS(20) TAU000530
87. C DIMENSION IALTE(150), AA(20,21), KK(5), D(5), PDELZ(20) TAU000540
88. C DIMENSION GGGA(200)
89. C DIMENSION STSUM(200)
90. C EQUIVALENCE (ARRAY(1,1), FX(1)), (ARRAY(1,2), SSELZ(1)), TAU000550
91. C 1 (ARRAY(1,3), RESA(1)), (ARRAY(1,4), GGGA(1))
92. C 100 FERRAT(5F5.1) TAU000570
93. C CALL TIC
94. C DO 1 K=1,200
95. C 1 FER(K)=0.
96. C ILHBP=0 TAU000580
97. C SREFC=0 TAU000590
98. C IREST=0
99. C IFIRST=0
100. C RSU=1.570 TAU000600
101. C IMOD=0 TAU000610
102. C DEL=.15 TAU000620
103. C D(2)=DEL TAU000630
104. C D(4)=DEL TAU000640
105. C KK(1)=1 TAU000650
106. C KK(3)=1 TAU000660
107. C KK(5)=1 TAU000670
108. C MCH=0
109. C D(1)=0.0
110. C D(3)=0.0
111. C D(5)=0.0
112. C KK(2)=0
113. C KK(4)=0
114. C DO 101 I=1,200
115. C RESA(I)=0.0
116. C FX(I)=0.0
117. C FZ(I)=0.0
118. C PDELZ(I)=0.0
119. C SSELZ(I)=0.0

```

```

120.      X(I)=0.0
121.      Z(I)=0.0
122.      SUM(I)=0.0
123.      DWGT(I)=0.0
124.      PCN(I)=0.0
125.      SUMA(I)=0.0
126.      STSUM(I)=0.0
127.      101 CONTINUE
128.      DO 102 I=1,20
129.      XS(I)=0.0
130.      REDEL(I)=0.0
131.      102 CONTINUE
132.      IIN=105
133.      IPUN=106
134.      IIGUT=108
135.      N4=4
136.      JTAPE=2
137.      KFXN=1
138.      C NOTE THAT THE LAST POLYGON MUST HAVE NUMBER 99
139.      L=39
140.      42 FORMAT(2F10.2)
141.      442 FORMAT(2F10.2/2I1)
142.      44 FORMAT(15,4F10.2)
143.      45 FORMAT(/26H LNB =,I4, 10H RHB =,F10.3,2X,5A4)
144.      47 FORMAT(/35H K FX(K) FZ(K) ANOMALY)
145.      51 FORMAT(/ K FX(K) FZ(K) ANOMALY CAL REF RESIDUAL)
146.      511 6S ANOMALY WEIGHT WGT DIFF WEIGHTEST,5X,'FILT')
147.      C RDENS = REFERENCE DENSITY FOR GRAVITY CALCULATIONS
148.      C RWGT = REFERENCE WEIGHT FOR MASS CALCULATIONS
149.      C RHMD = DIFFERENCE DENSITY FOR MODEL ADJUSTING = RHB(LBWR) - RHB(UPPER)
150.      READ(IIN,446) (LABEL(KU), KU=1,20)
151.      446 FORMAT(20A4)
152.      WRITE(IIGUT,447) (LABEL(KU), KU=1,20)
153.      447 FORMAT(1X,20A4)
154.      I=ISW(-2)
155.      IF (ISW(1).EQ.0) WRITE(JTAPE,446) (LABEL(KU), KU=1,20)
156.      DO 56 K=1,20
157.      56 IALTE(K)=ISW(K)
158.      IF (ISW(1).EQ.0) WRITE(JTAPE,419) (IALTE(K), K=1,20)
159.      419 FORMAT(80I1)
160.      READ(IIN,427) RDENS, RAGT, RHMD, REFX, FXI, DELFX, M, NFER, IMAX
161.      427 FORMAT(6F10.2, 110, 2I3)
162.      OUTPUT RDENS, RWGT, RHMD, REFX, FXI, DELFX, M, NFER, IMAX
163.      IZERB=0
164.      IF (ISW(1).EQ.0) WRITE(JTAPE,427) RDENS, RWGT, RHMD, REFX, FXI, DELFX, M
165.      *, IZERB, IMAX
166.      IF (NFER.LT.1) GO TO 5963
167.      OUTPUT WFER(K)
168.      DO 596 K=0, NFER
169.      WFER(K)=(NFER-K+1.)/(NFER+1.)
170.      WFER(-K)=WFER(K)
171.      KL=K
172.      WRITE(IIGUT,42) WFER(KL), WFER(K)
173.      C NOTE WE HAVE MADE A TRIANGULAR WEIGHTING FUNCTION
174.      5960 CONTINUE
175.      5963 CONTINUE
176.      DO 59 I=1, M
177.      SUM(I)=0.
178.      TEST(I)=0
179.      DSU(I)=0

```

TAU00680

TAU00690

TAU00700

TAU00710

TAU00720

TAU00730

TAU00740

TAU00750

TAU00760

TAU00770

TAU00780

TAU00800

TAU00810

TAU00830

TAU00840

TAU00850

TAU00860

TAU00870

TAU00880

TAU00890

TAU00900

TAU00910

TAU00920

TAU00930

TAU00940

TAU00950

TAU00960

TAU00980

TAU00990

TAU01010

TAU01020

TAU01030

TAU01040

180.	59	CONTINUE	TAU01050
181.		FX(KFXN)=FXI	TAU01060
182.		FZ(KFXN)=0.0	TAU01070
183.		ARRAY(KFXN,5)=0.	TAU01080
184.		K1=KFXN+1	TAU01090
185.		READ200,(9GA(I),I=1,M)	TAU01100
186.	200	FORMAT(5F10.1)	TAU01110
187.		DO 620 I=1,M	
188.		9GGA(I)=9GA(I)	
189.		IF(9GA(I).GE.899.) 9GGA(I)=0.	
190.	620	CONTINUE	
191.		IF(ISW(1).EQ.1) GO TO 7004	TAU01120
192.		WRITE(JTAPE,200) (9GA(I),I=1,M)	TAU01130
193.	7004	CONTINUE	
194.		IF (IS(2).EQ.0) GO TO 7021	
195.		READ201,(FZ(I),I=1,M)	TAU01150
196.	201	FORMAT(5F10.1)	TAU01170
197.		IF (ISW(1).EQ.1) GO TO 7005	TAU01180
198.		WRITE(JTAPE, 201) (FZ(I),I=1,M)	TAU01190
199.	7005	CONTINUE	TAU01200
200.		CC= 1.	
201.		IF (ISW(2).EQ.2) CC=1000.	
202.		DO 7034 I=1,M	TAU01210
203.		FZ(I)=-FZ(I)/CC	
204.		ARRAY(I,5)=FZ(I)*(-100.)	TAU01230
205.		IF (ISW(11).EQ.1) FZ(I)=0.	
206.	7034	CONTINUE	TAU01240
207.	7021	CONTINUE	TAU01250
208.		DO 451 K=K1,M	TAU01260
209.		FX(K)=FX(K-1)+DELEX	TAU01270
210.	4501	CONTINUE	TAU01280
211.	451	CONTINUE	TAU01290
212.		DO 36 K=KFXN,M	TAU01300
213.		SELZ(K) = 0.0	TAU01310
214.	46	CONTINUE	TAU01320
215.		DO 53J=KFXN,M	TAU01330
216.		IF (REFX-FX(J))53,21,53	TAU01340
217.	53	CONTINUE	TAU01350
218.	21	J=J	TAU01360
219.		REF9GA = 9GA(J)	TAU01370
220.		JREF=J	TAU01380
221.	60	CONTINUE	TAU01390
222.		READ(IIN,433) LNB,RH0RK,C0ME	
223.	433	FORMAT(I5,F10.3,5A4)	
224.		WRITE(IIOU,45) LNB,RH0RK,C0ME	
225.		IF (ISW(1).EQ.0) WRITE(JTAPE,433) LNB,RH0RK	TAU01430
226.		PH0=RH0*K*RDENS	TAU01440
227.	799	CONTINUE	
228.		I=1	TAU01450
229.	801	READ 442 ,XX,ZZ ,IC0DE ,IAL	TAU01460
230.		X(I)=XX	TAU01470
231.		Z(I)=ZZ	TAU01480
232.		IALTE(I)=IAL	TAU01490
233.		PRINT 7032,X(I),Z(I),IC0DE ,IAL	TAU01500
234.	7032	FORMAT(2X,2F11.3,3X,2I1)	TAU01510
235.		IF (ISW(1).EQ.1) GO TO 7008	TAU01520
236.		IF (IREST.EQ.1) GOTO 7008	
237.		WRITE(JTAPE,442)X(I),Z(I),IC0DE,IAL	TAU01530
238.	7008	CONTINUE	TAU01540
239.		N=I	TAU01550

240.		I=I+1	TAU01560
241.		IF (ICRDE-9) 801, 310, 801	TAU01570
242.	810	CONTINUE	TAU01580
243.		IF (IREST.EQ.1) GO TO 811	
244.		CALL REIG2 (X,Z,N,FX,M,SUM,RH0RK,TEST,DSU)	TAU01590
245.	811	CONTINUE	TAU01600
246.		IF (ISW(3).EQ.0) PRINT 47	TAU01610
247.	C		TAU01620
248.	C	FIELD POINT DB LOOP	TAU01630
249.	C		TAU01640
250.		DO 421 K=KFXN,M	TAU01650
251.		SDELZ=0.	TAU01660
252.	C		TAU01670
253.	C	POLYGON POINTS DB LOOP	TAU01680
254.	C		TAU01690
255.		DO 3004 I=1,N	TAU01700
256.		IDUM=1	TAU01710
257.	205	EXXX=X(I) - FX(K)	TAU01720
258.		ZFEF=Z(I) - FZ(K)	TAU01730
259.		CALL CAMP	TAU01740
260.	3004	CONTINUE	TAU01750
261.		IF (ILBOP.EQ.1) PDELZ(K)=RH0D*(13.34*SDELZ-PCBN(K)/RH0),G0T03005	TAU01760
262.		PDELZ(K)=13.34*RH0*SDELZ	TAU01770
263.	3005	CONTINUE	TAU01780
264.		SSELZ(K)=SSELZ(K)+PDELZ(K)	TAU01790
265.		IF (LNB-1) 4101,5001,4101	TAU01800
266.	5001	BGTD = BGA(K)+PDELZ(K)*(2.67-RH0RK)/RH0	TAU01810
267.	5006	PRINT 5007,K,FX(K),FZ(K),PDELZ(K),BGTD	TAU01820
268.	5007	FORMAT(15,4F10.2)	TAU01830
269.	C		
270.	C	TO WRITE COMPLETE BUGGUER ANOMALY ON TAPE	
271.	C		
272.		IF (ISW(1).EQ.1) GO TO 7009	
273.		WRITE(JTAPE,5007)K,FX(K),FZ(K),PDELZ(K),BGTD	
274.		GO TO 7009	
275.	4101	CONTINUE	TAU01850
276.		IF (ISW(3).EQ.1) GO TO 7022	TAU01860
277.		PRINT 44,K,FX(K),FZ(K),PDELZ(K),DSU(K)	TAU01870
278.	7022	CONTINUE	TAU01880
279.	5008	CONTINUE	TAU01890
280.		IF (ISW(9).EQ.1) GO TO 7009	TAU01900
281.		IF (ILBOP.EQ.1) GO TO 7009	TAU01910
282.		IF (ISW(1).EQ.1) GO TO 7009	TAU01920
283.		IF (IREST.EQ.1) G0T0 7009	
284.		WRITE(JTAPE,44)K,FX(K),FZ(K),PDELZ(K)	TAU01930
285.	7009	CONTINUE	TAU01940
286.	421	CONTINUE	TAU01950
287.		IF (ISW(4).EQ.1) GO TO 423	TAU01960
288.		CALL PLOTTER(PDELZ,M,BAR)	TAU01970
289.	423	CONTINUE	TAU01980
290.		IF (L-LNB) 60,430,60	TAU01990
291.	430	REFCOR = REF0GA-SSELZ(J)	TAU02000
292.		DO 422 K=KFXN,M	TAU02010
293.		SSELZ(K) = SSELZ(K) + REFCOR	TAU02020
294.	422	CONTINUE	TAU02030
295.		SREFC=SREFC+REFCOR	TAU02040
296.		IFES=0	TAU02050
297.		RSQD=RSQ	TAU02060
298.		RSQ=0	TAU02070
299.		RESF=0	

```

300. C
301. C CALCULATING THE RMS ERROR
302. C
303. DO 4422 K=KFXN,M
304. IF (OGA(K).GE.900) GOT0 4422
305. RESA(K)=SSELZ(K)-OGA(K)
306. 4422 CONTINUE
307. DO 4424 K=KFXN,M
308. IF (OGA(K).GE.900) GOT0 4424
309. IF (NFER.LT.1) FER(K)=RESA(K) ; GOT0 4029
310. FER(K)=0
311. WTFE=0
312. KSTART=K-NFER
313. KEND=K+NFER
314. DO 4020 KH=KSTART,KEND
315. IF (OGA(KH).GE.900. .OR. KH.LT.1 .OR. KH.GT.M) GOT0 4020
316. FER(K)=FER(K)+RESA(KH)*WFER(K=KH)
317. WTFE=WTFE+WFER(K=KH)
318. 4020 CONTINUE
319. FER(K)=FER(K)/WTFE
320. 4029 CONTINUE
321. IRES=IRES+1
322. RESF=FER(K)**2+RESF
323. RSQ=RESA(K)**2+RSQ
324. 4424 CONTINUE
325. RESF=SQRT(RESF/IRES)
326. RSQ=SQRT(RSQ/IRES)
327. WRITE(OUTPUT,4425) IMBD,RSQ,IRES,RESF
328. 4425 FORMAT(2X,IMBD=I5, RMS ERROR=F10.5, NUMBER OF PRINTS=I5,
329. * , FILTERED RMS ERROR=F10.5)
330. IF (IMBD.GT.IMAX) GOT0 438
331. IF (ISW(13).EQ.1 .AND. IFIRST.EQ.0) GOT0 439
332. 4423 CONTINUE
333. IFIRST=1
334. IF (ISW(10).EQ.1 .AND. IMBD.LE.1) GOT0 4427
335. IF (ISW(6).EQ.0 .OR. IMBD.GT.IMAX .OR. ((RSQD-.5).LT.RSQ .AND.
336. * (IMBD.GT.1))) GOT0 438
337. C BRANCHING OUT OF MODEL ALTERING PART OF M PROGRAM
338. 4427 CONTINUE
339. DO 4425 LQ=1,MCH
340. DO 4425 LQ2=1,MCH+1
341. 4425 AA(LQ,LQ2)=0
342. X(1)=X(N)
343. Z(1)=Z(N)
344. K=JREF
345. MCH=M
346. C
347. C COMPUTATION OF D/DZ FOR THE REFERENCE POINT
348. C
349. DO 7650 I=1,N=1
350. IF (IALTE(I).EQ.0) GOT0 7650
351. SDELZ=0
352. DO 7640 IDUM=1,5
353. II=IDUM
354. ARG=X(I+1)-X(I=1)
355. EXXX=X(I+KK(II))-FX(K)
356. ZEEE=Z(I+KK(II))-FZ(K)+D(II)*SIGN(1.,ARG)
357. CALL COMP
358. 7640 CONTINUE
359. MCH=MCH+1

```

TAU02080  
TAU02090  
TAU02100

TAU02130

TAU02110  
TAU02120

TAU02140

TAU02150  
TAU02160

TAU02170  
TAU02180

TAU02210

TAU02220

TAU02230

TAU02240

TAU02250

TAU02260

TAU02270

TAU02280

TAU02290

TAU02300

TAU02310

TAU02320

TAU02330

TAU02340

TAU02350

TAU02360

TAU02370

TAU02380

TAU02390

TAU02400

TAU02410

```

360.      RFDEL(MCH)=13.34*RHOD*SDELZ                      TAU02420
361.      7650  CONTINUE                                     TAU02430
362.      4701  FORMAT(1X,I2,10G10.3)                     TAU02440
363.      KCODE=0                                           TAU02450
364.      IF(ISW(7).EQ.1) WRITE(IIBUT,4701) KCODE,K, (RFDEL(KD),KD=1,MCH) TAU02460
365.      C                                               TAU02470
366.      C   STRING THE PLD VALUES OF PDELZ               TAU02480
367.      C                                               TAU02490
368.      C   IF(ILBOP.EQ.1) GO TO 434                     TAU02500
369.      DO 432 K=KFXN,M                                     TAU02510
370.      432   PCYN(K)=PDELZ(K)                             TAU02520
371.      GO TO 436                                           TAU02530
372.      434   DO 435 K=KFXN,M                               TAU02540
373.      435   SSELZ(K)=SSELZ(K)+PDELZ(K)                 TAU02550
374.      C   SSELZ(K) IS NOW THE SUMMED VALUE OF ALL POLYGON CONTRIBUTIONS
375.      C   EXCEPT THAT DUE TO THE CHANGED PART OF THE MODEL
376.      SREFC=SREFC-PDELZ(JREF)                            TAU02560
377.      436  CONTINUE                                     TAU02570
378.      ILBOP=1                                           TAU02580
379.      IMOD=IMOD+1                                       TAU02590
380.      CC                                               TAU02600
381.      C   FIELD POINT DO L68P                           TAU02610
382.      C                                               TAU02620
383.      DO 7850 K=KFXN,M                                   TAU02630
384.      IF(BOA(K).GE.900) GO TO 7850
385.      IF(K.EQ.JREF) GO TO 7850
386.      MCH=0
387.      C                                               TAU02650
388.      C   COMPUTING D/DZ FOR EACH CHANGEABLE POLY POINT TAU02660
389.      C                                               TAU02670
390.      DO 7830 I=1,N-1                                     TAU02680
391.      IF(IALTE(I).EQ.0) GO TO 7830                       TAU02690
392.      SDELZ=0.                                           TAU02700
393.      DO 7820 II=1,5                                       TAU02710
394.      ARG=X(I+1)-X(I-1)                                   TAU02720
395.      EXXX=X(I+KK(II))-FX(K)                             TAU02730
396.      ZEEE=Z(I+KK(II))-FZ(K)+D(II)*SIGN(1.,ARG)         TAU02740
397.      IDUM=II                                           TAU02750
398.      CALL COMP                                           TAU02760
399.      7820  CONTINUE                                     TAU02770
400.      MCH=MCH+1                                           TAU02780
401.      DDELZ(MCH)=13.34 *RHOD*SDELZ-RFDEL(MCH)          TAU02790
402.      7830  CONTINUE                                     TAU02800
403.      KCODE=1                                           TAU02810
404.      IF(ISW(7).EQ.1) WRITE(IIBUT,4701) KCODE,K, (DDELZ(KD),KD=1,MCH) TAU02820
405.      C   NOW ADD THE CONTRIBUTION TO THE NORMAL EQUATION TAU02830
406.      C                                               TAU02840
407.      DO 7835 II=1,MCH                                    TAU02850
408.      DO 7834 IP=1,MCH                                    TAU02860
409.      AA(II,IP)=AA(II,IP)+DDELZ(IP)*DDELZ(II)          TAU02870
410.      7834  CONTINUE                                     TAU02880
411.      AA(II,MCH+1)=AA(II,MCH+1)+FER(K)*DDELZ(II)      TAU02890
412.      7835  CONTINUE                                     TAU02900
413.      7850  CONTINUE                                     TAU02910
414.      C   WE HAVE NOW FINISHED SETTING UP THE NORMAL EQUATIONS TAU02920
415.      EPS=1.E-20                                         TAU02930
416.      INDIC=1                                             TAU02940
417.      NRC=20                                             TAU02950
418.      C                                               TAU02960
419.      C   NOTE THAT NRC IS THE NUMBER OF COLUMNS IN THE MATRIX AA TAU02970

```

```

420. C MP1=MCH+1 TAU02990
421. IF (ISW(7).EQ.1) OUTPUT 'NORMAL EQUATIONS'; WRITE (IISUT,4540 TAU03000
422. * ), (MP1, (AA(II,JJ), JJ=1, MP1), II=1, MCH) TAU03010
423. 4540 FORMAT (NG10.3) TAU03020
424. CALL SIMUL(MCH, AA, XS, EPS, INDIC, NRC, DETER, TAU03030
425. IF (ISW(7).EQ.1) OUTPUT DETER, (XS(IW), IW=1, MCH) TAU03040
426. C CALCULATE THE NEW VALUES OF THE POLYGON POINTS TAU03050
427. MCH=0 TAU03060
428. DO 7860 K=1, N TAU03070
429. IF (IALTE(K).EQ.0) GO TO 7850 TAU03080
430. MCH=MCH+1 TAU03090
431. Z(K)=Z(K)+XS(MCH)*2*DEL TAU03100
432. IF (Z(K).LT.0.2) Z(K)=.2 TAU03110
433. 7860 CONTINUE TAU03120
434. IF (ISW(7).EQ.1) OUTPUT 'NEW POLY POINTS'; OUTPUT, (Z(K), K=1, N) TAU03130
435. GO TO 411 TAU03140
436. CONTINUE TAU03150
437. IF (ILBPP.EQ.0) GO TO 439 TAU03160
438. C MAKING THE WEIGHT CALCULATION FOR THE MODIFIED POLYGON TAU03170
439. C AA IS JUST A GARBAGE ARRAY TAU03180
440. CALL WEIG2(X, Z, N, FX, M, AA, 1, AA, DWGT) TAU03190
441. DO 441 K=KFXN, M TAU03200
442. IF (IREST.EQ.1) SUM(K)=STSUM(K)
443. STSUM(K)=SUM(K)
444. C STORING THE VALUE OF SUM TO USE IN FUTURE CALCULATIONW
445. 441 SUM(K)=SUM(K)+(DWGT(K)-DSU(K)/RH0RK) *RH0D TAU03210
446. C DWGT(K) IS BEING USED FOR TEMPORARY STORAGE
447. OUTPUT 'NEW POLY POINTS -FINAL VERSION' TAU03220
448. WRITE (IISUT,440) (X(K), Z(K), K=1, N) TAU03230
449. 440 FORMAT (2X, 2F10.2) TAU03240
450. IF (IREST.EQ.1) GOTO 4041
451. IF (ISW(1).EQ.0) WRITE (JTAPE, 42) (X(K), Z(K), K=1, N-1);
452. * WRITE (JTAPE, 442) X(N), Z(N), IC0DE
453. 4041 CONTINUE
454. 439 PRINT 51 TAU03290
455. DO 7000 K=KFXN, M
456. DWGT(K)=SUM(K)-RWGT TAU03310
457. SSSSS=SSELZ(K)-SREFC TAU03320
458. IF (ISW(12).EQ.1)
459. * DWGT(K)=DWGT(K)+ARRAY(K,5)*2.67
460. PRINT 52, K, FX(K), FZ(K), SSSSS, SSELZ(K), RESA(K), 0GA(K) TAU03330
461. 1, SUM(K), DWGT(K), TEST(K), FLR(K) TAU03340
462. 52 FORMAT (15.6F10.2, F15.0, F16.0, F16.0, F6.1)
463. 483 CONTINUE TAU03350
464. IF (ISW(1).EQ.1) GO TO 7000 TAU03370
465. IF (IREST.EQ.1) GOTO 7000
466. WRITE (JTAPE, 52) K, FX(K), FZ(K), SSSSS, SSELZ(K), RESA(K), 0GA(K) TAU03380
467. 1, SUM(K), DWGT(K) TAU03390
468. 7000 CONTINUE TAU03400
469. CALL PLOTTER(DWGT, M, BAR) TAU03410
470. CONTINUE TAU03420
471. 4220 IF (ISA(1).EQ.1) GO TO 7013 TAU03430
472. END FILE JTAPE TAU03440
473. 7013 CONTINUE TAU03450
474. 610 CONTINUE TAU03470
475. IF (ISW(5).EQ.1) N4=5
476. ARRAY(200,1)=ARRAY(M,1) TAU03490
477. CALL PLOTA(LN0, ARRAY, 200, N4, M, 0.0, 1, FDUM, FDUM) TAU03530
478. OUTPUT ' 1= THEORETICAL GRAVITY IN MILIGALS' TAU03540
479.

```

```

480.      OUTPUT : 2= DIFFERENCE BETWEEN THEORETICAL AND OBSERVED GRAVITY; TAU03550
481.      OUTPUT : 3= OBSERVED GRAVITY; TAU03560
482.      OUTPUT : 4= ELEVATION IN 10 S OF METERS; TAU03570
483.      IF (ISW(13).EQ.1 .AND. IFIRST.EQ.0) GO TO 4423
484.      999 CONTINUE TAU03580
485.      C STORING THE VALUE OF X
486.      IF (ISW(6) .NE.1) GO TO 7011
487.      READ(IIN,433,END=7011) LNB,RHNEW
488.      IF (LNB.NE.99) GO TO 7011
489.      IREST=1
490.      IMED=0
491.      RS=1.E70
492.      DO 630 K=KFXN,M
493.      630 SSELZ(K)=SSELZ(K)-FDELZ(K)
494.      C SSELZ(K) IS NOW THE SUMMED VALUE OF ALL POLYGON CONTRIBUTIONS
495.      C EXCEPT THAT DUE TO THE CHANGED PART OF THE MODEL
496.      OUTPUT : NOW RECALCULATING THE MODEL USING NEW VARIABLE POLYPOINT
497.      *S*
498.      GO TO 799
499.      C IREST IS SET EQUAL TO 1 TO INDICATE THAT
500.      C WE ARE READING AN ADDITIONAL SET OF POINTS FOR THE LAST
501.      C POLYGON TO SEE THE EFFECT OF USING DIFFERENT
502.      C VARIABLE POLYGON POINTS
503.      7011 CONTINUE
504.      CALL TBC(TIME)
505.      OUTPUT TIME
506.      STOP TAU03620
507.      SUBROUTINE COMP TAU03630
508.      RR=EXXX**2+ZEEE**2 TAU03640
509.      IF (EXXX) 210,240,280 TAU03650
510.      210 IF (ZEEE) 220,230,270 TAU03660
511.      220 THETB=ATAN(ZEEE/EXXX)+3.1415927 TAU03670
512.      GO TO 301 TAU03680
513.      230 THETB=ATAN(ZEEE/EXXX)+3.1415927 TAU03690
514.      GO TO 301 TAU03700
515.      240 IF (ZEEE) 250,260,270 TAU03710
516.      250 THETB=-1.5707963 TAU03720
517.      GO TO 301 TAU03730
518.      260 THETB=0. TAU03740
519.      GO TO 301 TAU03750
520.      270 THETB=1.5707963 TAU03760
521.      GO TO 301 TAU03770
522.      280 THETB=ATAN(ZEEE/EXXX) TAU03780
523.      301 IF (IDUM=1) 3001,3002,3001 TAU03790
524.      3001 CHECK=EXX*ZEEE-ZEE*EXXX TAU03800
525.      IF (CHECK) 320,310,320 TAU03810
526.      310 DELZ=0. TAU03820
527.      GO TO 401 TAU03830
528.      320 OMEGA=THETA-THETB TAU03840
529.      IF (OMEGA) 3201,3202,3202 TAU03850
530.      3202 IF (OMEGA+3.1415927) 330,330,340 TAU03860
531.      3201 IF (OMEGA+3.1415927) 340,330,330 TAU03870
532.      330 DTHET=OMEGA TAU03880
533.      GO TO 370 TAU03890
534.      340 IF (OMEGA) 351,360,360 TAU03900
535.      351 DTHET=OMEGA+6.2831853 TAU03910
536.      GO TO 370 TAU03920
537.      360 DTHET=OMEGA-6.2831853 TAU03930
538.      370 A=CHECK/((EXXX-EXX)**2+(ZEEE-ZEE)**2) TAU03940
539.      B=(EXXX-EXX)*DTHET TAU03950

```

```
540.      C=).5*(ZEEF-ZEE)*ALOG(RR/R)
541.      DELZ=1*(B+C)
542.      401 SDELZ=SDELZ+DELZ
543.      3002 EXX=EXXX
544.      ZFE=ZFEF
545.      R=RR
546.      THETA=THETB
547.      RETURN
548.      END
```

```
TAU03960
TAU03970
TAU03980
TAU03990
TAU04000
TAU04010
TAU04020
TAU04030
TAU04040
```

[illegible]

00000	BAB	00001	LABEL	00015	C9ME	0001A	FX	0001A	ARRAY	000E2	SSELZ
001AA	RESA	00272	8GA	00702	FZ	004CA	PDELZ	00593	XO	00593	X
0045B	ZD	0045C	PCGN	00724	8GA	007EC	TEST	008B4	DSU	0097C	SUM
00444	DAGT	008BC	PCGN	00BD4	SUMW	00C9C	RDEFL	00CB0	XS	00CC4	IATLE
005A	AA	00CEB	KK	00F03	O	00F08	DDELZ	00F1C	STSUM	00FE4	K
005A	ILBBP	00CEB	SREFC	00FE7	IEST	00FER	IFIRST	00FE9	RSQ	00FEA	IM9D
00TEB	DEL	00FEC	MCH	00FED	I	00FF4	IIN	00FF6	IPUN	00FF7	I19UT
00EL	JN	00FER	JTAPE	00FF3	KFXN	00FF4	L	00FF5	KU	00FF6	RDENS
00EF7	RAGT	00FER	RHSD	00FF9	REFX	00FFA	FXI	00FFB	DELFX	00FFC	M
00FED	NFER	00FEF	IMAX	00FEF	IZER0	01000	KL	01001	K1	01002	CC
01003	J	01004	95F9GA	01005	JRF5	01006	LN9	01007	R0RK	01008	RH9
01007	XX	01004	ZZ	01006	ICDEE	0100C	IAL	0100D	N	0100E	SDELZ
010CF	IDUM	01010	XXX	01011	ZEEF	01012	BGTD	01013	REFCND	01014	IRES
01015	PSQLD	01015	RESF	01017	WTFE	01018	KSTART	01019	KEND	0101A	KH
0101B	LQ	0101C	LOG	0101D	II	0101E	ARG	0101F	KCODE	01020	KD
01021	IP	01022	EFS	01023	INDIC	01024	NRC	01025	PI1	01026	JJ
01027	DEFER	01028	IW	01029	SSSSS	0102A	FDUM	0102B	RHNEW	0102C	TIME
0102D	C5XP	0102E	RR	0102F	TMETB	01030	CHECK	01031	EXX	01032	ZEE
01033	DELZ	01034	RMGA	01035	TMETA	01036	DTHET	01037	A	01038	B
01039	C	0103A	R	0103B							

00000 FER 00000 WFER

00720 CAMP

ALG	ATAN	SIGN	SGRT
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	42
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	51	51	51
52	52	52	52
53	53	53	53
54	54	54	54
55	55	55	55
56	56	56	56
57	57	57	57
58	58	58	58
59	59	59	59
60	60	60	60
61	61	61	61
62	62	62	62
63	63	63	63
64	64	64	64
65	65	65	65
66	66	66	66
67	67	67	67
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	76	76	76
77	77	77	77
78	78	78	78
79	79	79	79
80	80	80	80
81	81	81	81
82	82	82	82
83	83	83	83
84	84	84	84
85	85	85	85
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	94
95	95	95	95
96	96	96	96
97	97	97	97
98	98	98	98
99	99	99	99
100	100	100	100

ISW	PLCTA	PLAYER
F:102	F:103	F:104
9ALGG	9ATAN1	9PCORDEE
916DATA	91PLUSA	91TR

```

TIC      T9C      W192      F:101
F:106    F:108    V:D9      M:9C
9RCDWRIT 9ENDFILE 9ENC19L 9INITIAL
9READ     9SETUP9 9SORT   9ST99

```

F:101  
M:8C  
GINITIAL  
9STAD

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	1946	0079A
CONSTANTS:	35	00023
LOCAL VARIABLES:	4155	0103B
TEMPS:	8	00008
TOTAL PROGRAM:	6144	01800 (PLUS BLANK COMMON)

```

1.      SUBROUTINE ALTD(ELEV,IDEF,HEIGT,KK)
2.      C   VERSION OF 13 JANUARY 1971
3.      C   SUBROUTINE ALTD,   RETURNS VALUE OF HEIGT (NEGATIVE BELOW SEA
4.      C   LEVEL).   KK RETURNS = 9 IF NEW DATA RECORD SHOULD BE
5.      C   READ,   OTHERWISE KK = 0.
6.      KK=0
7.      A=ABS(ELEV)
8.      IF(A=0.004)100,100,200
9.      C   ELEV = ZERO
10.     C   CHECKING DEPTH
11.     100   IF(IDEF)110,110,130
12.     110   KK=9
13.           GO TO 990
14.     120   HEIGT=ELEV
15.           GO TO 990
16.     130   HEIGT=-IDEF
17.           GO TO 990
18.     C   ELEV NOT ZERO
19.     200   IF(ELEV)210,100,210
20.     210   HEIGT=ELEV
21.           GO TO 990
22.     990   RETURN
23.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
A	R	SCALR	00001 V	1	ABS	R	SPR8G	INTRIN	---	ALTD	R	SPR8G	00000 P	---
ALTD	R	SCALR	00000 V	1	ELEV	R	SCALR	*00002 V	DUMMY	HEIGT	R	SCALR	*00004 V	DUMMY
DEP	I	SCALR	*00003 V	DUMMY	KK	I	SCALR	*00005 V	DUMMY					

LABEL	HEX L8C	LABEL	HEX L8C
100 0000E	---	110 00010	---
990 0002C	---		

## LOCAL VARIABLES (2 WORDS):

0000C ALTD 00001 A

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

0000C ALTD

## INTRINSIC SUBPROGRAMS USED:

ABS

## EXTERNAL SUBPROGRAMS REQUIRED:

9IT8R 9SETUPN

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	33	00021
CONSTANTS:	1	00001
LOCAL VARIABLES:	2	00002
TEMPS:	5	00005
TOTAL PROGRAM:	41	00029

```

1.      SUBROUTINE ANBV2(ZZ,ZHT,XX,KGCA,KGM8,KGYR,KGHM,K8GDA,XX,YY,INIT,
2.      1 DATA,IDECL)
3.      C *      24 JULY 1974      - TO ADD HORIZ AND VERT ANNOTATION
4.      C
5.      C
6.      C
7.      C SUBROUTINE ANBV2 ANNOTATES PLOTTED POINT WITH DATA VALUE, TIME, OR DATE AND
8.      C TIME
9.      C
10.     C SSW(3) UP TO ANNOTATE ONLY AT CHANGE OF DATE
11.     C SSW(7) UP TO ANNOTATE ON LEFT SIDE OF TRACK
12.     C SSW(11) UP TO ANNOTATE ALTERNATELY ON LEFT AND RIGHT SIDES OF TRACK
13.     C SSW(18) 0 TO HAVE ANNOTATIONS AT RIGHT ANGLES TO INCREMENTAL TRACK
14.     C              1 TO ANNOTATE HORIZONTALLY
15.     C              2 TO ANNOTATE VERTICALLY
16.     C              3 ON TRACK HEADING 091 TO 269, TO INVERT ANNOTATION
17.     C              4 TO ANNOTATE EITHER HORIZ OR VERT DEPENDING UPON DIRECTION
18.     C
19.     C      USES CALCOMP SUBROUTINES AND ISW
20.     C
21.     C      DIMENSION HM(4), BNTH(2), DAY(2)
22.     C      HGT=0.07*ZHT
23.     C      IF(INIT) 80,80,85
24.     C      THETA IS THE INCREMENTAL TREND OF TRACK
25.     C      80 AX=XX-X8LD
26.     C      AY=YY-Y8LD
27.     C      OPTION TO PLOT VALUES HORIZONTALLY OR VERTICALLY
28.     C      IF(ISW(18) .NE. 1) GO TO 110
29.     C      THATA=C
30.     C      IF(ABS(AY - C*08)) 400,58,58
31.     C      110 IF(ISW(18) .NE. 2) GO TO 111
32.     C      THATA=1.57079
33.     C      IF(ABS(AX - C*08)) 400,58,58
34.     C      111 CONTINUE
35.     C      IF(ISW(18).EG.4) GO TO 29
36.     C      DETERMINE ANGLE ALONG WHICH TO ANNOTATE
37.     C      IF(AY) 50,51,51
38.     C      50 THATA=ABS(AY/AX)
39.     C      THATA=1.57079+ATAN(THATA)
40.     C      IF(AX.GT.C) THATA=-THATA
41.     C      GO TO 59
42.     C      51 THATA=ABS(AX/AY)
43.     C      THATA=ATAN(THATA)
44.     C      IF(AX.GT.C) THATA=-THATA
45.     C      59 IF(ISW(18).EG.3.AND.THATA.GT.1.57079) THATA=THATA-3.14159
46.     C      IF(ISW(18).EG.3.AND.THATA.LT.-1.57079) THATA=THATA+3.14159
47.     C      GO TO 58
48.     C      29 BX=ABS(XX-X8LD)
49.     C      BY=ABS(YY-Y8LD)
50.     C      IF(BX-BY) 36,36,30
51.     C      ANNOTATE VERTICALLY
52.     C      30 THATA=1.57079
53.     C      GO TO 58
54.     C      ANNOTATE HORIZONTALLY
55.     C      36 THATA=C.0
56.     C      58 THETA=THATA*57.29578
57.     C      CHECK IF DISTANCE INCREMENT ALONG TRACK FROM LAST DATA POINT IS SUFFICIENT
58.     C      THAT NEXT ANNOTATION DOES NOT OVERPRINT LAST
59.     C      TAX=SGRT(AX*AX+AY*AY)

```

```

60.      IF (ISW(11)) 71,71,70
61.      70 TAX=TAX+O.C4*ZHT
62.      71 TAX=TAX+O.C8*ZHT
63.      C IF TAX IS NEGATIVE, INCREMENT IS INSUFFICIENT TO ANNOTATE, RETURN
64.      IF (TAX) 400,92,92
65.      92 IF (ISW(11).NE.1) GO TO 60
66.      61 KBLUNT=KBLUNT+1
67.      ISIDE=MOD(KBLUNT,2)
68.      IF (ISIDE) 93,93,94
69.      60 IF (ISW(7)) 94,94,93
70.      93 BFSET=-O.C34
71.      GO TO 95
72.      94 BFSET=O.C08
73.      95 XXT=BFSET*COS(THATA)
74.      YTT=BFSET*SIN(THATA)
75.      A=C.14*ZHT*SIN(THATA)
76.      B=C.14*ZHT*COS(THATA)
77.      C=C.C7*ZHT*SIN(THATA)
78.      D=C.C7*ZHT*COS(THATA)
79.      XT=XX+XXT
80.      YT=YY+YTT
81.      100 IF (NX.NE.1) GO TO 320
82.      C CHANGE HOUR AND MINUTE FORMAT FROM (14) TO (411) FORMAT SO TO PRINT FOUR
83.      C DIGITS
84.      HM(1)=KGHM/1000
85.      HM(2)=(KGHM-HM(1)*1000.)/100.
86.      KHM1=HM(2)
87.      HM(2)=KHM1
88.      HM(3)=(KGHM-(HM(1)*1000.++HM(2)*100.))/10.
89.      KHM1=HM(3)
90.      HM(3)=KHM1
91.      HM(4)=KGHM-(HM(1)*1000.++HM(2)*100.++HM(3)*10.)
92.      C CHANGE DAY AND MONTH FORMATS FROM (12) TO (211) SO TO PRINT TWO DIGITS.
93.      DAY(1)=KGDA/10
94.      DAY(2)=KGDA-(DAY(1)*10.)
95.      BNTH(1)=KGM0/10
96.      BNTH(2)=KGM0-(BNTH(1)*10.)
97.      C ANNOTATE DATA POINT
98.      280 IF (KGDA.NE.K0GDA.AND.ISW(3).EG.1) GO TO 291
99.      IF (KGDA-K0GDA) 291,290,291
100.     291 CALL NUMBER(XT,YT,HGT,DAY(1),THETA,-1)
101.     XT=XT+D
102.     YT=YT+C
103.     CALL NUMBER(XT,YT,HGT,DAY(2),THETA,-1)
104.     XT=XT+B
105.     YT=YT+A
106.     CALL NUMBER(XT,YT,HGT,BNTH(1),THETA,-1)
107.     XT=XT+D
108.     YT=YT+C
109.     CALL NUMBER(XT,YT,HGT,BNTH(2),THETA,-1)
110.     XT=XT+B
111.     YT=YT+A
112.     YEAR=KGYR
113.     CALL NUMBER(XT,YT,HGT,YEAR,THETA,-1)
114.     IF (ISW(3).EG.1) GO TO 340
115.     XT=XT+(2.O*B)
116.     YT=YT+(2.O*A)
117.     290 IF (ISW(3).EG.1) GO TO 340
118.     CALL NUMBER(XT,YT,HGT,HM(1),THETA,-1)
119.     XT=XT+D

```

```

120.      YT=YT+C
121.      CALL NUMBER(XT,YT,HGT,HM(2),THETA,-1)
122.      XT=XT+D
123.      YT=YT+C
124.      CALL NUMBER(XT,YT,HGT,HM(3),THETA,-1)
125.      XT=XT+D
126.      YT=YT+C
127.      CALL NUMBER(XT,YT,HGT,HM(4),THETA,-1)
128.      GO TO 340
129.      320 CALL NUMBER (XT, YT, HGT, DATA, THETA, IDEC)
130.  C  RETURN FEN TO DATA POINT
131.      340 CALL PLOT (XX,YY,3)
132.      342 XOLD=XX
133.           YOLD=YY
134.      400 RETURN
135.      85 KBLNT=0
136.           THATA=C.
137.           THETA=C.
138.           XOLD=C.
139.           YOLD=C.
140.      GO TO 60
141.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	00018 V	1	ABS	R	SPR0G	INTRIN	1	ANB2	R	SPR0G	00000 P	1	AX	R	SCALR	00000 V	1
ANB2	R	SCALR	00000 V	1	ATAN	R	SPR0G	INTRIN	1	AX	R	SCALR	00000 V	1	BX	R	SCALR	00000 V	1
AY	R	SCALR	00000 V	1	B	R	SCALR	00019 V	1	CBS	R	SPR0G	00007 V	2	CBS	R	SPR0G	00007 V	2
BY	R	SCALR	00000 V	1	C	R	SCALR	0001A V	1	DAY	R	ARRAY	00007 V	2	DAY	R	ARRAY	00007 V	2
D	R	SCALR	00000 V	1	DATA	R	SCALR	0001A V	1	DECE	R	SCALR	00002C V	DUMMY	DECE	R	SCALR	00002C V	DUMMY
HGT	R	SCALR	00009 V	1	HM	R	ARRAY	00001 V	4	ISW	R	SCALR	00002C V	DUMMY	ISW	R	SCALR	00002C V	DUMMY
INIT	R	SCALR	00002A V	DUMMY	ISICE	R	SCALR	00014 V	1	ISW	R	SCALR	00002C V	DUMMY	ISW	R	SCALR	00002C V	DUMMY
KGDA	R	SCALR	000023 V	DUMMY	KGDA	R	SCALR	00026 V	1	KGDA	R	SCALR	00002C V	DUMMY	KGDA	R	SCALR	00002C V	DUMMY
KGVR	R	SCALR	000025 V	DUMMY	KHPT	R	SCALR	0001E V	1	KHPT	R	SCALR	00002C V	DUMMY	KHPT	R	SCALR	00002C V	DUMMY
KBUNT	R	SCALR	000013 V	1	MBC	R	SCALR	0001E V	1	NUMBER	R	SCALR	00002C V	DUMMY	NUMBER	R	SCALR	00002C V	DUMMY
NX	R	SCALR	000022 V	DUMMY	OFSET	R	SCALR	00015 V	1	OFSET	R	SCALR	00002C V	DUMMY	OFSET	R	SCALR	00002C V	DUMMY
PL0T	R	SCALR	000022 V	DUMMY	SIN	R	SCALR	00015 V	1	SIN	R	SCALR	00002C V	DUMMY	SIN	R	SCALR	00002C V	DUMMY
TAX	R	SCALR	000012 V	1	THATA	R	SCALR	0000E V	1	THATA	R	SCALR	00002C V	DUMMY	THATA	R	SCALR	00002C V	DUMMY
XBLD	R	SCALR	000009 V	1	XT	R	SCALR	0000E V	1	XT	R	SCALR	00002C V	DUMMY	XT	R	SCALR	00002C V	DUMMY
XYT	R	SCALR	000016 V	1	YEAR	R	SCALR	0000E V	1	YEAR	R	SCALR	00002C V	DUMMY	YEAR	R	SCALR	00002C V	DUMMY
YT	R	SCALR	00001D V	1	YY	R	SCALR	0000E V	1	YY	R	SCALR	00002C V	DUMMY	YY	R	SCALR	00002C V	DUMMY
ZHT	R	SCALR	000021 V	DUMMY	ZZ	R	SCALR	0000E V	1	ZZ	R	SCALR	00002C V	DUMMY	ZZ	R	SCALR	00002C V	DUMMY

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
29	0006C	30	00078	36	0007B	50	0003C	51	0004A
59	00056	60	000A6	61	0009D	70	0008D	71	00091
85	001C3	92	00098	93	000AB	94	000AE	95	00080
110	00028	111	00035	280	0011E	290	00179	291	00129
340	001B9	342	001BE	400	001C2				

## LOCAL VARIABLES (32 WORDS):

CCCC ANB2	00001 HM	00007 DAY	00009 HGT	0000A AX
CCCC XBLD	0000C AY	0000E THATA	0000F BX	00010 BY
CCCC THETA	00012 TAX	00014 ISIDE	00015 BFSEY	00016 XXT
CCCC YTT	00018 A	0001A C	0001B D	0001C XT
CCCC YT	0001E KHM			

## BLANK COMMON (C WORDS)

## ENTRY PRINTS:

CCCC ANB2

## INTRINSIC SUBPROGRAMS USED:

ABS ATAN COS M0D SIN SQRT

## EXTERNAL SUBPROGRAMS REQUIRED:

ISW ALPBER PL0T 9ATAN1 9COS 9IT0R 9RT0I 9SETUPN  
9SIN 9SCT

HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	462	001CE
CONSTANTS:	23	00017
LOCAL VARIABLES:	32	00020
TEMPS:	17	00011
TOTAL PROGRAM:	534	00216

```

1.      SUBROUTINE ANBV3(XX,YY,DEPT,AMAG)
2.      C      VERSION OF 15 DEC 1971,  ADD SIZE VARIABLE AND CHANG
3.      C      DEPTH LIMITS FOR SHALLOW EPICENTERS
4.      C      SUBROUTINE ANBV3,  TO MAKE VARIABLE SIZED SYMBOLS FOR
5.      C      EPICENTER DATA DEPENDING UPON DEPTH AND MAGNITUDE
6.      C
7.      C      DEPT = DEPTH IN KM
8.      C      AMAG = MAGNITUDE  (MAXIMUM IS 7.5)
9.      C
10.     C
11.     DATA ISTRT/0/
12.     IF(ISTRT)15,5,15
13.     5  OUTPUT 'ANBV3, VER 15 DEC 1971'
14.     C  SIZE =1.0
15.     SIZE =2.0
16.     OUTPUT SIZE
17.     ISTRT = 1
18.     C
19.     C      END OF INITIALIZATION
20.     C
21.     15  IF(DEPT= 70.0)20,20,22
22.     20  INTEG=1
23.     GO TO 50
24.     22  IF(DEPT=150.0)24,24,26
25.     24  INTEG=2
26.     GO TO 50
27.     26  IF(DEPT=300.0)28,28,30
28.     28  INTEG=5
29.     GO TO 50
30.     30  IF(DEPT=500.0)32,32,34
31.     32  INTEG=12
32.     GO TO 50
33.     34  INTEG=0
34.     GO TO 50
35.     50  IF(AMAG=4.5)52,52,54
36.     52  HF=1.0
37.     GO TO 100
38.     54  IF(AMAG=5.5)56,56,58
39.     56  HF=2.0
40.     GO TO 100
41.     58  IF(AMAG=6.5)60,60,62
42.     60  HF=3.0
43.     GO TO 100
44.     62  HF=4.0
45.     GO TO 100
46.     100 HGT=0.07*HF*SIZE
47.     CALL SYMBOL(XX,YY,HGT,INTEG,0.0,-1)
48.     RETURN
49.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
APAG	R	SCALR	*00009	V DUMMY	AN8V3	R	SCALR	00000	V	AN8V3	R	SCALR	00000	P
DEPT	R	SCALR	*00008	V DUMMY	HF	R	SCALR	00004	V	HGT	R	SCALR	00005	V
INTEG	I	SCALR	00003	V	ISTR	I	SCALR	00001	V	SIZE	R	SCALR	00002	V
SYMBOL	SPR6G	EXTERN		1	XX	R	SCALR	*00006	V DUMMY	YY	R	SCALR	*00007	V DUMMY

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
5	CCCC05	15	00022	20	CC025	22	CC028
28	CCC31	30	00034	32	CC037	34	CC03A
54	CCC43	56	CC046	58	CC049	60	CC04C

## LOCAL VARIABLES (6 WORDS):

00000	AN8V3	00001	ISTR	00002	SIZE	00003	INTEG	00004	HF	00005	HGT
-------	-------	-------	------	-------	------	-------	-------	-------	----	-------	-----

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

00000 AN8V3

## EXTERNAL SUBPROGRAMS REQUIRED:

SYMBOL	F:IC8	SENDI8L	9I8DATA	9PRINT	9SETUPN
--------	-------	---------	---------	--------	---------

## HIGHEST ERROR SEVERITY: 0 (N9 ERRORS)

GENERATED CODE:	DEC WORDS	HEX WORDS
CONSTANTS:	95	CC05F
LOCAL VARIABLES:	14	CC00E
TEMP:	6	CC006
TEMP:	5	CC005
TOTAL PROGRAM:	120	CC078

```
1.      SUBROUTINE AREAK(DLAT,CLONG,IAKEY)
2.      C      VERSION OF 1 DECEMBER 1971
3.      C
4.      C      VERSION OF 18 OCT 1971, DUMMY ROUTINE
5.      C
6.      C      SUBROUTINE AREAK, GIVES VALUE TO CODE IAKEY WHICH MAY
7.      C      SUBSEQUENTLY BE USED IN SORTING GSUM RECORDS INTO A
8.      C      SO DESIGNATED BY DIFFERENT VALUES FOR IAKEY
9.      C
10.     C
11.     IAKEY = 0
12.     RETURN
13.     END
```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
DLONG	----	----	-----	-----	DLAT	----	----	-----	-----	DLAT	----	----	-----	-----
AREAK	----	SPRNG	0C000 F	-----	AREAK	----	----	-----	-----	AREAK	----	----	-----	-----
		LAUSED	*CC002 V	DUMMY	IAKEY	----	I SCALR	*000C3	DUMMY	IAKEY	----	I SCALR	*000C3	DUMMY

LOCAL VARIABLES (1 WORD):

0C00C ARAK

BLANK COMMON (0 WORDS)

ENTRY POINTS:

0C00C ARAK

EXTERNAL SUBPROGRAMS REQUIRED:

9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	9	CCCC9
CONSTANTS:	0	CCCC0
LOCAL VARIABLES:	1	CCCC1
TEMPS:	4	CCCC4
TOTAL PROGRAM:	14	CCCC14

```
1.          SUBROUTINE CALSC(A,B,C,D,SC,CC)
2.  C  SUBROUTINE CALSC, DETERMINES SIN AND COS OF ANGLE
3.  C      OF TILT OF DIGITIZED MAP
4.          R=SGRT((C-A)**2+(D-B)**2)
5.          SC=(D-B)/R
6.          CC=(C-A)/R
7.          RETURN
8.          END
```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	*C0002	V DUMMY	B	R	SCALR	*C0003	V DUMMY	C	R	SCALR	*00004	V DUMMY
CALSC	R	SCALR	C000C	V 1	CALSC	R	SCALR	00000	P	CC	R	SCALR	*00007	V DUMMY
C	R	SCALR	*C0005	V DUMMY	R	R	SCALR	00001	V	SC	R	SCALR	*00006	V DUMMY
SCRT	R	SPR8G	INTRIN											

## LOCAL VARIABLES (2 WORDS):

C000C CALSC C0001 R

## BLANK COMMON (C WORDS)

## ENTRY POINTS:

C000C CALSC

## INTRINSIC SUBPROGRAMS USED:

SCRT

## EXTERNAL SUBPROGRAMS REQUIRED:

9SETUPN 9SGRT

## HIGHEST ERROR SEVERITY: C (NO ERRORS)

DEC WORDS	HEX WORDS
28	C001C
C	C000C
2	C0002
8	C0008
38	C0026

TOTAL PROGRAM:

```

1.      SUBROUTINE CDATE (IDA1,IM01,IYR1,IHM1,
2.      1 IDA2,IM02,IYR2,IHM2,TIMD)
3.      C
4.      C * 2 DEC 1970 /2200 -- S.ABBOT
5.      C H-F ASA BASIC FORTRAN (EXTENDED)
6.      C MODIFIED FOR SIGMA 7 -- 20 DEC 71
7.      C
8.      C * PURPOSE: COMPARES TWO DATES AND RETURNS THE TIME
9.      C DIFFERENCE IN DECIMAL HOURS (TIMD) ;
10.     C 'TIMD' WILL BE NEGATIVE IF DATE 1 IS AFTER DATE 2.
11.     C
12.     C * EQUIVALENT TO SUBR. 'CDATR' EXCEPT THAT THE HOUR-MINUTE
13.     C ARGUMENTS ARE SUPPLIED AS INTEGER NUMBERS.
14.     C
15.     C * THERE ARE NO DATE LIMITS FOR INPUT DATA
16.     C
17.     C      AHM1 = IHM1
18.     C      AHM2 = IHM2
19.     C      CALL NCH (IDA1,IM01,IYR1,AHM1,ID1,T1)
20.     C      CALL NCH (IDA2,IM02,IYR2,AHM2,ID2,T2)
21.     C
22.     C * CALCULATE TIME DIFFERENCE IN DECIMAL HOURS
23.     C
24.     C      TIMD = (ID2 - ID1)
25.     C      TIMD = TIMD * 24.
26.     C      TIMD = TIMD + (T2-T1)
27.     C
28.     C      RETURN
29.     C      END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
AM1	R	SCALR	00001 V	1	AM2	R	SCALR	00002 V	1	AM2	R	SCALR	00002 V	1
CDATE	R	SCALR	00000 V	1	IDA1	I	SCALR	00007 V	DUMMY	IDA2	I	SCALR	00008 V	DUMMY
ID1	I	SCALR	00003 V	1	ID2	I	SCALR	00005 V	DUMMY	IM1	I	SCALR	0000A V	DUMMY
IM2	I	SCALR	0000E V	DUMMY	IM1	I	SCALR	00008 V	DUMMY	IM2	I	SCALR	0000C V	DUMMY
IYR1	I	SCALR	00009 V	DUMMY	IYR2	I	SCALR	0000D V	DUMMY	NDH	SPR8G	EXTERN		
TIME	R	SCALR	0000F V	DUMMY	T1	R	SCALR	00004 V	1	T2	R	SCALR	00006 V	1

## LOCAL VARIABLES (7 WORDS):

C000C CDATE	00001 AM1	00002 AHM2	00003 ID1	00004 T1	00005 ID2
C0006 T2					

BLANK COMP6N (0 WORDS)

ENTRY POINTS:

C000C CDATE

EXTERNAL SUBPROGRAMS REQUIRED:

NDH 9108 9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	46	C002E
CONSTANTS:	1	C0001
LOCAL VARIABLES:	7	C0007
TEMPS:	10	C000A
TOTAL PROGRAM:	64	C0040

```

1.      SUBROUTINE CHGMT(KDA,KM0,KYR,KHM,KTZ,KGDA,
2.      1  KGM0,KGYR,KGHM,NTZ)
3.      DIMENSION M0DAY(12)
4.      C  SUBROUTINE CHGMT TO DETERMINE GMT DATE AND TIME
5.      C  FROM LOCAL TIME
6.      C  THE SIGN OF THE TIME ZONE DIFFERENCE IS TO GO
7.      C  FROM GMT TIME TO THE LOCAL TIME.  THUS IF
8.      C  GMT = 1800, AND LOCAL = 1400, KTZ = -04.
9.      C
10.     M0DAY(1)=31
11.     M0DAY(2)=28
12.     M0DAY(3)=31
13.     M0DAY(4)=30
14.     M0DAY(5)=31
15.     M0DAY(6)=30
16.     M0DAY(7)=31
17.     M0DAY(8)=31
18.     M0DAY(9)=30
19.     M0DAY(10)=31
20.     M0DAY(11)=30
21.     M0DAY(12)=31
22.     KGHM=KHM-(KTZ*100)
23.     IF(KGHM)110,126,128
24.     110 KGHM=2400+(KHM-(KTZ*100))
25.     KGDA=KDA+1
26.     A=KYR
27.     B=KYR/4
28.     A=A/4.0
29.     IF(A-B)112,122,112
30.     112 IF(KGDA)114,114,120
31.     114 KGM0=KM0-1
32.     IF(KGM0)116,118,116
33.     116 KGYR=KYR
34.     KGDA=M0DAY(KM0-1)
35.     GO TO 150
36.     118 KGM0=12
37.     KGDA=M0DAY(KGM0)
38.     KGYR=KYR-1
39.     GO TO 150
40.     120 KGM0=KM0
41.     KGYR=KYR
42.     GO TO 150
43.     122 IF(KM0-3)112,123,112
44.     123 IF(KDA-1)112,124,112
45.     124 KGDA=M0DAY(KM0-1)+1
46.     KGM0=KM0-1
47.     KGYR=KYR
48.     GO TO 150
49.     126 KGDA=KDA
50.     KGM0=KM0
51.     KGYR=KYR
52.     GO TO 150
53.     128 IF(KGHM-2400)126,131,130
54.     131 KGHM=C000
55.     132 KGDA=KDA+1
56.     A=KYR
57.     B=KYR/4
58.     A=A/4.0

```

```
60.      134  IF (KGDA=MODAY(KM0))136,136,138
61.      136  KGM0=KM0
62.      KGYR=KYR
63.      G0 T0 150
64.      138  KGDA=1
65.      KGM0=KM0+1
66.      IF (KGM0=13)140,142,140
67.      140  KGYR=KYR
68.      G0 T0 150
69.      142  KGM0=1
70.      KGYR=KYR+1
71.      G0 T0 150
72.      144  IF (KM0=2)134,146,134
73.      146  IF (KGDA=29)134,136,138
74.      130  KK=KTZ*100
75.      KGHM=(KHM-KK)-2400
76.      G0 T0 132
77.      150  ATZ=-KTZ
78.      RETURN
79.      END
```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	0C00D V	I
CHGMT	R	SCALR	0C00C V	I
KGHM	I	SCALR	*CC018 V	DUMMY
KHM	I	SCALR	*CC013 V	DUMMY
KTZ	I	SCALR	*CC014 V	DUMMY
KTZ	I	SCALR	*CC019 V	DUMMY
B	R	SCALR	COCCE V	I
KDA	I	SCALR	*C0C10 V	DUMMY
KGPB	I	SCALR	*CCC16 V	DUMMY
KK	I	SCALR	*C0C13 V	DUMMY
KYR	I	SCALR	*C0C12 V	DUMMY
NAME <th>TYPE</th> <th>CLASS</th> <th>HEX LBC</th> <th>DEC WORDS</th>	TYPE	CLASS	HEX LBC	DEC WORDS
CHGMT	R	SCALR	COCCE V	I
KGDA	I	SCALR	*C0C10 V	DUMMY
KGYR	I	SCALR	*CCC16 V	DUMMY
KMB	I	SCALR	*C0C13 V	DUMMY
M0DAY	I	SCALR	*C0C12 V	DUMMY
NAME <th>TYPE</th> <th>CLASS</th> <th>HEX LBC</th> <th>DEC WORDS</th>	TYPE	CLASS	HEX LBC	DEC WORDS
CHGMT	R	SPRG6	00000 P	P
KGDA	I	SCALR	*00015 V	DUMMY
KGYR	I	SCALR	*00017 V	DUMMY
KMB	I	SCALR	*00011 V	DUMMY
M0DAY	I	ARRAY	00001 V	12

LABEL	HEX	L6C	LABEL	HEX	L6C	LABEL	HEX	L6C	LABEL	HEX	L6C	LABEL	HEX	L6C	LABEL	HEX	L6C
110	CC02D	112	00043	CC045	114	00043	CC045	116	0004A	118	00050	120	00059	122	0005E	124	00061
122	CC05E	123	00061	CC064	124	00061	CC064	126	CC06E	128	00075	130	000AD	131	00079	132	0007B
131	CC079	132	0007B	CC08B	134	0007B	CC08B	136	CC08F	138	00094	140	0009C	141	CC09F	142	CC0A5
142	CC09F	144	000A5	CC0AB	146	000A5	CC0AB	150	CC0B5								

LOCAL VARIABLES (16 WORDS):

00000	CHGMT	00001	M0DAY	00000	A	00000	E	B	00000	F	KK
-------	-------	-------	-------	-------	---	-------	---	---	-------	---	----

BLANK CARRIER (0 WORDS)

### ENTRY POINTS:

CCCC CHGT

**EXTERNAL SUBPROGRAMS REQUIRED:**

SITØR 9SETUFA

HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC	HEX
	WORDS	WORDS
GENERATED CODE:	185	C0CB9
CONSTANTS:	3	CC003
LOCAL VARIABLES:	16	CCC1C
TEMPS:	11	CC00B
TOTAL PROGRAM:	215	CC0D7

1.		SUBROUTINE C08RR(X,Y,RLONG,RLAT,ITST,JTST)	G3DC3810
2.	C	CHANGES LAT AND LONG TO TRANSVERSE MERCATOR AND VICA VERSA	G3DC3820
3.	C	VERSION OF JUNE 1972	G3DC3830
4.	C	MODIFIED BY BRUCE SIMON	G3DC3840
5.		IIN=105	G3DC3850
6.		IIOUT=108	G3DC3860
7.		IF(JTST) 2,2,15	G3DC3870
8.	2	IF(ITST)1,1,102	G3DC3880
9.			G3DC3890
10.	1	CONTINUE	G3DC3900
11.		ITST=1	G3DC3910
12.		E2=.676865799E-2	G3DC3920
13.		DTR=3.141592653/180.	G3DC3930
14.		RTD=1./DTR	G3DC3940
15.		ARCRD=360C.*RTD*30.713114	G3DC3950
16.		AA=6378206.4	G3DC3960
17.		AD=111132.089	G3DC3970
18.		A=AD/DTR	G3DC3980
19.		B=16216.944	G3DC3990
20.		C=17.20937	G3DC4000
21.		D=0.02273	G3DC4010
22.		E=0.000033	G3DC4020
23.		G1=1./25.523932E-10	G3DC4030
24.		AS=.484813681E-5	G3DC4040
25.	101	CONTINUE	G3DC4050
26.		READ(IIN,601) CMD,CMM,PZD,PZM,ISR,XZ,YZ	G3DC4060
27.	601	FORMAT(F4.0,F6.3,F4.0,F6.3,I10,2F10.0)	G3DC4070
28.		WRITE(IIOUT,607) CMD,CMM,PZD,PZM,ISR,XZ,YZ	G3DC4080
29.	607	FORMAT(' CM=',F4.0,' DEG',F6.3,' MIN.',PZ=' ,F4.0,' DEG',F6.3,' MIN',G3DC4090	
30.	1,	ISR=' ,I6,' XZ=' ,F6.0,' , YZ=' ,F6.0)	G3DC4100
31.		XZ=XZ*1000.	G3DC4110
32.		YZ=YZ*1000.	G3DC4120
33.		PZ=PZD+PZM/60.	G3DC4130
34.		CM=CMD+CMM/60.	G3DC4140
35.		RCM=-CM*DTR	G3DC4150
36.		RPZ=PZ*DTR	G3DC4160
37.		IF(ISR) 5,6,5	G3DC4170
38.	5	R=FL0AT(ISR-1)/FL0AT(ISR)	G3DC4180
39.		G0 T0 7	G3DC4190
40.	6	R=1.	G3DC4200
41.	7	CONTINUE	G3DC4210
42.		EL0=A*RPZ-B*SIN(2.*RPZ)+C*SIN(4.*RPZ)-D*SIN(6.*RPZ)	G3DC4220
43.	1	+E*SIN(8.*RPZ)	G3DC4230
44.		EL0=EL0*R	G3DC4240
45.		RM=AD-566.05*C0S(2.*RPZ)+1.2*C0S(4.*RPZ)	G3DC4250
46.		RM=RM*R*RTD	G3DC4260
47.		RETURN	G3DC4270
48.	102	CONTINUE	G3DC4280
49.		IFLG=2	G3DC4290
50.		X=X*1000.	G3DC4300
51.		Y=Y*1000.	G3DC4310
52.		XP=X-XZ	G3DC4320
53.		YP=Y-YZ	G3DC4330
54.		P1=RPZ+YP/RM	G3DC4340
55.	10	CONTINUE	G3DC4350
56.		ELN=(A*P1-B*SIN(2.*P1)+C*SIN(4.*P1)-D*SIN(6.*P1)	G3DC4360
57.	1	+E*SIN(8.*P1))*R	G3DC4370
58.		OY=ARCRD*R/SGRT((1.-E2*(SIN(P1))**2)**3)	G3DC4380
59.		YN=ELN-EL0	G3DC4390
		OPN=(YP-YN)/OY	

60.		P1=F1+DPN	G3DC4400
61.		IFLG=IFLG-1	G3DC4410
62.		IF (IFLG) 11,11,10	G3DC4420
63.	11	CONTINUE	G3DC4430
64.	CC		G3DC4440
65.	C	P1 IS NOW THE TABULAR LATITUDE	G3DC4450
66.	CC		G3DC4460
67.		T=(1.-E2*(SIN(P1))**2)	G3DC4470
68.		C=TAN(P1)*T**2/G1	G3DC4480
69.	C	WE ARE NOW CALCULATING G RHO Z FACTOR	G3DC4490
70.		SG1=(XF*(XP**3)*(T/R)**2/242.436946E+12)/R	G3DC4500
71.		DELP=((SG1)**2*C/3600.)*DTR	G3DC4510
72.		P1=P1-DELP	G3DC4520
73.	C	P1 IS NOW THE TRUE LATITUDE	G3DC4530
74.		T=(1.-E2*(SIN(P1))**2)	G3DC4540
75.		RN=AA/SGRT(T)	G3DC4550
76.		DELL1=SIN(SG1/RN)/COS(P1)	G3DC4560
77.		DELLR=ARSIN(DELL1)	G3DC4570
78.		RLONG=-DELLR+RCM	G3DC4580
79.		RLONG=-RLONG	G3DC4590
80.		RLAT=P1	G3DC4600
81.		X=X/1000.	G3DC4610
82.		Y=Y/1000.	G3DC4620
83.		RETURN	G3DC4630
84.	15	DELLR=RCM+RLONG	G3DC4640
85.		CELL1=SIN(DELLR)	G3DC4650
86.		F1=RLAT	G3DC4660
87.		F2=P1	G3DC4670
88.		T=1.-E2*(SIN(P2))**2	G3DC4680
89.		RN=AA/SGRT(T)	G3DC4690
90.		SG1=RN*ARSIN(COS(P2)*DELL1)	G3DC4700
91.		DO 85 I=1,3	G3DC4710
92.		C=TAN(P2)*T**2/G1	G3DC4720
93.		DELP=((SG1)**2*C/3600.)*DTR	G3DC4730
94.	85	P2=P1+DELP	G3DC4740
95.		P1=P2	G3DC4750
96.		ELN=(A*P1-B*SIN(2.*P1)+C*SIN(4.*P1)-D*SIN(6.*P1)	G3DC4760
97.	1	+E*SIN(8.*P1))*R	G3DC4770
98.		Y=ELN-EL0+Y2	G3DC4780
99.		SG=R*SG1	G3DC4790
100.		T=(1.-E2*(SIN(P1))**2)	G3DC4800
101.		XF=SG*SG**3*(T/R)**2/242.436946E+12	G3DC4810
102.		X=XP+XZ	G3DC4820
103.		X=X/1000.	G3DC4830
104.		Y=Y/1000.	G3DC4840
105.		RETURN	G3DC4850
106.		END	G3DC4860

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	HEX LBC	DEC WORDS
A	R	SCALR	00009	1	AA	R	SCALR	00007	1	AD	R	SCALR	00008	1		
ARCRD	R	SCALR	00006	1	ARSIN	R	SPR0G	0000B	1	AS	R	SCALR	0000F	1		
B	R	SCALR	0000A	1	CM	R	SCALR	0000B	1	CM	R	SCALR	00018	1		
CMD	R	SCALR	00010	1	CMP	R	SCALR	00011	1	C08RR	R	SCALR	00000	1		
C08RR	R	SPR0G	00000	1	C08	R	SPR0G	00000	1	D	R	SCALR	0000C	1		
DELLR	R	SCALR	0002B	1	DELL1	R	SCALR	0002A	1	DELP	R	SCALR	00028	1		
DFN	R	SCALR	00025	1	DTR	R	SCALR	00004	1	DY	R	SCALR	00023	1		
E	R	SCALR	0002D	1	ELN	R	SCALR	00022	1	EL0	R	SCALR	0001C	1		
E2	R	SCALR	00003	1	FL0AT	R	SPR0G	00001	1	I	I	SCALR	0002D	1		
IFLG	I	SCALR	0001E	1	IIN	I	SCALR	000C3	1	IIBUT	I	SCALR	00002	1		
ISR	I	SCALR	00014	1	ITST	I	SCALR	*00033	DUMMY	JTST	I	SCALR	*00034	DUMMY		
PZ	R	SCALR	00017	1	PZD	R	SCALR	00012	1	PZM	R	SCALR	00013	1		
P1	R	SCALR	00021	1	P2	R	SCALR	0002C	1	Q1	R	SCALR	0000E	1		
R	R	SCALR	0001B	1	RCM	R	SCALR	00019	1	RLAT	R	SCALR	*00032	DUMMY		
RL0NG	R	SCALR	*00031	DUMMY	RM	R	SCALR	0001D	1	RN	R	SCALR	00029	1		
RPZ	R	SCALR	0001A	1	RTD	R	SCALR	00005	1	SG	R	SCALR	0002E	1		
SG1	R	SCALR	00027	1	SIN	R	SPR0G	00000	1	SGRT	R	SPR0G	INTRIN	DUMMY		
T	R	SCALR	00026	1	TAN	R	SPR0G	INTRIN	1	X	R	SCALR	*0002F	DUMMY		
XP	R	SCALR	0001F	1	XZ	R	SCALR	00015	1	Y	R	SCALR	*00030	DUMMY		
YN	R	SCALR	00024	1	YP	R	SCALR	00020	1	YZ	R	SCALR	00016	1		

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
1	00011	5	00083	6	0008C	7	0008E
11	00108	85	00183	101	00033	102	0008E
607	00052						

# LOCAL VARIABLES (47 WORDS):

C000C	C08RR	00001	IIN
C0006	ARCRD	00007	AA
C000C	D	0000C	E
C0012	PZD	00013	PZM
C0018	CM	00019	RCM
C001E	IFLG	0001F	XP
C0024	YN	00025	DFN
C002A	DELL1	0002B	DELLR

BLANK COMMON (0 WORDS)

## ENTRY PRINTS:

C000C C08RR

## INTRINSIC SUBPROGRAMS USED:

ARSIN	C6S	FL0AT	SIN	SGRT	TAN
00004	0000A	00010	00016	0001C	00022
00005	0000B	00011	00017	0001D	00023
00006	0000C	00012	00018	0001E	00024
00007	0000D	00013	00019	0001F	00025
00008	0000E	00014	0001A	00020	00026
00009	0000F	00015	0001B	00021	00027
00010	00010	00016	0001C	00022	00028
00011	00011	00017	0001D	00023	00029
00012	00012	00018	0001E	00024	0002A
00013	00013	00019	0001F	00025	0002B
00014	00014	0001A	00020	00026	0002C
00015	00015	0001B	00021	00027	0002D
00016	00016	0001C	00022	00028	0002E
00017	00017	0001D	00023	00029	
00018	00018	0001E	00024		
00019	00019	0001F	00025		
0001A	0001A	00020	00026		
0001B	0001B	00021	00027		
0001C	0001C	00022	00028		
0001D	0001D	00023	00029		
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E	0001E	00024			
0001F	0001F	00025			
0001A	0001A	00020			
0001B	0001B	00021			
0001C	0001C	00022			
0001D	0001D	00023			
0001E					

## EXTERNAL SUBPROGRAMS REQUIRED:

F:101	F:102	F:103	F:104	F:105	F:106	F:108	9ASIN
9BCDREAD	9BCCWRIT	9C0S	9I0DATA	9IT0R	9SETUPN	9SIN	9SQRT
9TAN							

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	457	001C9
CONSTANTS:	23	00017
LOCAL VARIABLES:	47	0002F
TEMPS:	11	0000B
TOTAL PROGRAM:	538	0021A

```

1.      SUBROUTINE DISAZ(ALAT,ALON,BLAT,BLON,IFRD,AZMTH,BAKAZ,DISKM,DISDG)
2.      C THIS SUBROUTINE CALCULATES THE DISTANCE FROM A POINT A TO A POINT B
3.      C AN ALSO THE AZIMUTH CLOCKWISE FOR THE NORTH FROM A TO B
4.      C ALAT AND ALON ARE THE COORDINETS OF POINT A
5.      C BLAT AND BLON ARE THE COORDINETS OF POINT B
6.      C REMEMBER:::::
7.      C THE POSITION COORDINATES
8.      C - IS SOUTH AND WEST AND + IS NORTH AND EAST
9.      C
10.     C IRAD IS AN OPTION TO INPUT EITHER DEGREES OR RADIANS FOR THE
11.     C IF IRAD = C STUFF IN DGRES ; 1 IF IN RADIANS
12.     C THIS PROGRAM CAME FOR CALTECH( ) AND A WAS REVISED FOR THE MIT
13.     C IBM1194 COMPUTER BY JOHN FAIRBORN, IT THEN WAS REVISED FOR THE
14.     C TH IBM 360 AT MIT BY JACK WOLFE . NOW IT HAS BEEN REVISED FOR THE
15.     C SIGMA -7 COMPUTER AT WOODS HOLE (BY JACK WOLFE). THE INEFFICIENCY
16.     C OF STORAGE ALLOCATION IS DUE TO THE MOST RECENT PROGRAMMERS
17.     C (JACK WOLFE) LAZINES AND NEGLECT TO CLEAN UP ALL THE JUNK
18.     C DIMENSION TH(2),PHI(2),XDEG(2),DIST(2),AZ(2),AZINV(2)
19.     C TAN(THETA)=SIN(THETA)/COS(THETA)
20.     C I=1
21.     C K=2
22.     C TH(1)=ALAT
23.     C TH(2)=BLAT
24.     C PHI(1)=ALON
25.     C PHI(2)=BLON
26.     C IF(IFRD) 30,31,30
27. 31    C 32 J=1,2
28.     C TH(J)=TH(J)/57.2957795
29. 32    C PHI(J)=PHI(J)/57.2957995
30. 30    C CONTINUE
31.     C THG = ATAN(.99328 *TAN(TH(K)))
32.     C D = SIN(PHI(K))
33.     C E = -COS(PHI(K))
34.     C F = -COS(THG )
35.     C A = F+E
36.     C B = COS(THG )*D
37.     C C = SIN(THG )
38.     C G = -C+E
39.     C H = C*D
40.     C THG = ATAN(.99328 *TAN(TH(I)))
41.     C C1 = SIN(PHI(I))
42.     C E1 = -COS(PHI(I))
43.     C F1 = -COS(THG )
44.     C C1 = SIN(THG )
45.     C A1 = F1+E1
46.     C B1 = -F1*D1
47.     C G1 = -C1+E1
48.     C H1 = C1*D1
49.     C SC = A*A1 + B*B1 + C*C1
50.     C SD = SGRT(((A -A1)**2 + (B-B1)**2 + (C-C1)**2)*((A +A1)**2 + (B+B1)
51.     C 1**2 + (C+C1)**2)/4.0)
52.     C XDEG (I) = ATAN(SD/SC)*57.2957795
53.     C IF (SC) 1, 2, 2
54.     C XDEG (I) = XDEG(I) + 180.0
55. 1    C SS = ((A1-D)**2 + (B1-E)**2 + C1**2 - 2.0)
56. 2    C SC = ((A1-G)**2 + (B1-H)**2 + (C1-F)**2-2.0)
57.     C AZ(I) = ATAN(SS/SC)*57.2957795
58.     C IF (SS) 3,4,5
59. 3    C IF (SC) 6, 7, 7

```

```

0090
0100
0110
0120
0130
0140
0150
0160
0240
0250
0260
0270
0280
0290
0300
0310
0320
0330
0340
0350
0360
0370
0380
0390
0400
0410
0420

```

```

60.      6      AZ(I) = AZ(I) + 180.0      0430
61.      G0 T0 4      0440
62.      7      AZ(I) = AZ(I) + 360.0      0450
63.      G0 T0 4      0460
64.      5      IF (SC) 8, 4, 4      0470
65.      8      AZ(I) = AZ(I) + 180.0      0480
66.      4      SS = ((A-D1)**2 + (B-E1)**2 + C**2 - 2.0)      0490
67.      SC = ((A-G1)**2 + (B-H1)**2 + (C-F1)**2-2.0)      0500
68.      AZINV(I) = ATAN(SS/SC) *57.2957795      0510
69.      IF (SS) 13, 14, 15      0520
70.      13      IF (SC) 16, 17, 17      0530
71.      16      AZINV(I) = AZINV(I) + 180.0      0540
72.      G0 T0 14      0550
73.      17      AZINV(I) = AZINV(I) + 360.0      0560
74.      G0 T0 14      0570
75.      15      IF (SC) 18, 14, 14      0580
76.      18      AZINV(I) = AZINV(I) + 180.0      0590
77.      14      EL = 6.72267C02/.993277329E+03      0600
78.      EC = .672267C02E-02      0610
79.      E1 = 1.0+ EL      0620
80.      AL = TAN(TH(I))/(      E1*TAN(TH(K))) + EC *SQRT((E1+(TAN(TH(I)))
81.      1**2)/(E1 +(TAN(TH(K)))**2))      0630
82.      DL = PHI(I) - PHI(K)      0640
83.      A12 = ATAN(SIN(DL)/(      (AL -COS(DL))*SIN(TH(K))))      0650
84.      E0 = EL*((COS(TH(K)) *COS(A12))**2 +(SIN(TH(K)))**2)      0670
85.      E02 = E0**2      0680
86.      E03 = E0**3      0690
87.      C0 = 1.0+ E0/4.0+ 3.0*E02/64.0+ 5.0*E03/256.0      0700
88.      C2 = -E0/8.0+ E02/32.0+15.0*E03/1024.0      0710
89.      C4 = -E02/256.0+ 3.0*E03/1024.0      0720
90.      TH2 = TH(K)*2.0      0730
91.      TH2I = TH(I)*2.0      0740
92.      V1=EXP(.230259      E+01*(.380544      E+01*(.732368 E-3)*COS(TH2)
93.      1 +(.6175 E-6)*COS(2.0*TH2) -(.7E-9)*COS(3.0*TH2) ))      0760
94.      V2 = EXP(.230259      E+01*(.380544      E+01*(.732368 E-3)*COS(TH2I)
95.      1 +(.6175E-6)*COS(2.0*TH2I) -(.7E-9)*COS(3.0*TH2I)))      0770
96.      Z1 = V1*(1.0-      EC)*SIN(TH(K))      0780
97.      Z2 = V2*(1.0-      EC)*SIN(TH(I))      0790
98.      X2 = V2*COS(TH(I))*COS(DL)      0800
99.      Y2 = V2*COS(TH(I))*SIN(DL)      0810
100.      U1 = ATAN(TAN(TH(K))/(SQRT(1.0+E0)*COS(A12))      0820
101.      U2 = ATAN((V1*SIN(TH(K)) + (1.0+ E0)*(Z2-Z1))/(SQRT(1.0+E0)*      0830
102.      1X2*COS(A12) - Y2*SIN(TH(K))*SIN(A12)))      0840
103.      B0 = V1*SQRT(1.0+EL*(COS(TH(K))*COS(A12))**2)/(1.0+E0)      0850
104.      DIST(I)=B0*(C0*(U2-U1)+C2*(SIN(2.0*U2) -SIN(2.0*U1)) +C4*(SIN(4.0*
105.      1U2)- SIN(4.0*U1)) )      0860
106.      DIST(I) = ABS(DIST(I))      0870
107.      TEST = DIST(I) - 111.0*XDEG(I)      0880
108.      IF (ABS(TEST)-100.0) 25,301,301      0890
109.      301      U2 = U2 + .314159265E-01      0900
110.      DIST(I)=B0*(C0*(U2-U1)+C2*(SIN(2.0*U2) -SIN(2.0*U1)) +C4*(SIN(4.0*
111.      1U2)- SIN(4.0*U1)) )      0910
112.      25      CONTINUE      0920
113.      DISDG=XDEG(I)      0930
114.      DISKM=DIST(I)      0940
115.      AZMTH=AZ(I)
116.      BAKAZ=AZINV(I)
117.      C      BAKAZ IS THE AZZIMUTH FROM POINT B TO A
118.      RETURN
119.      END

```

NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS
A	R	SCALR	00014 V	1	ABS	R	SPR8G	INTRIN	1	AL	R	SCALR	00026 V	1
ALAT	R	SCALR	*0003B V	DUMMY	ALBN	R	SCALR	*0003C V	DUMMY	ATAN	R	SPR8G	INTRIN	1
AZ	R	ARRAY	00009 V	2	AZ1V	R	ARRAY	000CB V	2	AZMTH	R	SCALR	*00040 V	DUMMY
B	R	SCALR	00010 V	1	A12	R	SCALR	00028 V	1	B	R	SCALR	00015 V	1
BAKAZ	R	SCALR	*00041 V	DUMMY	BLAT	R	SCALR	*0003D V	DUMMY	BLON	R	SCALR	*0003E V	DUMMY
B0	R	SCALR	00039 V	1	B1	R	SCALR	0001E V	1	C	R	SCALR	00016 V	1
C0	R	SCALR	0002C V	1	C9S	R	SPR8G	INTRIN	1	C1	R	SCALR	0001C V	1
C2	R	SCALR	0002D V	1	C4	R	SCALR	0002E V	1	D	R	SCALR	00011 V	1
CISAZ	R	SPR8G	00000 F	1	D1S2	R	SCALR	0000C V	1	DISDG	R	SCALR	*00043 V	DUMMY
DISKM	R	SCALR	*00042 V	DUMMY	DIST	R	ARRAY	000C7 V	2	DL	R	SCALR	00027 V	1
D1	R	SCALR	00019 V	1	E	R	SCALR	00012 V	1	EC	R	SCALR	00025 V	1
EEL	R	SCALR	00024 V	1	E0	R	SCALR	00029 V	1	E02	R	SCALR	0002A V	1
E03	R	SCALR	0002B V	1	EXP	R	SPR8G	INTRIN	1	E1	R	SCALR	0001A V	1
F0	R	SCALR	00013 V	1	F1	R	SCALR	0001B V	1	G	R	SCALR	00017 V	1
G1	R	SCALR	0001F V	1	H	R	SCALR	00018 V	1	H1	R	SCALR	00020 V	1
I	I	SCALR	0000D V	1	IFRC	I	SCALR	*0003F V	DUMMY	J	I	SCALR	0000F V	1
K	I	SCALR	0000E V	1	PHI	R	ARRAY	000C3 V	2	SC	R	SCALR	00021 V	1
SD	R	SCALR	00022 V	1	SIN	R	SPR8G	INTRIN	1	SQRT	R	SPR8G	INTRIN	1
SS	R	SCALR	00023 V	1	TAN	R	SPR8G	000CD P	1	TEST	R	SCALR	0003A V	1
TH	R	ARRAY	00001 V	2	THETA	R	SCALR	*00044 V	DUMMY	THETA	R	UNUSED		
THG	R	SCALR	00010 V	1	TH2	R	SCALR	0002F V	1	TH21	R	SCALR	00030 V	1
U1	R	SCALR	00037 V	1	U2	R	SCALR	00038 V	1	V1	R	SCALR	00031 V	1
V2	R	SCALR	00032 V	1	XDEG	R	ARRAY	00005 V	2	X2	R	SCALR	00035 V	1
Y2	R	SCALR	00036 V	1	Z1	R	SCALR	00033 V	1	Z2	R	SCALR	00034 V	1

[illegible]

## LOCAL VARIABLES (59 WORDS):

[illegible]

BLANK CIPHER (C WORDS)

ENTRY POINTS:

00000 DISA Z

## LOCAL SUBPROGRAMS DEFINED:

0000D TAN

## INTRINSIC SUBPROGRAMS USED:

ABS          ATAN          COS          EXP          SIN          SQRT

## EXTERNAL SUBPROGRAMS REQUIRED:

9ATAN1      9COS      9EXP      9SETLFA      9SETUP1      9SIN      9SQRT

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	651	0028B
CONSTANTS:	28	0001C
LOCAL VARIABLES:	59	0003B
TEMPS:	20	00014
	-----	-----
TOTAL PROGRAM:	758	002F6

```

1.      FUNCTION DMTOR (KD, AM)
2.      C
3.      C
4.      C
5.      C
6.      C
7.      C
8.      C
9.      DMTOR = KD
10.     A = DMTOR + (AM/60.0)
11.     DMTOR = A * 1.745329E-2
12.     RETURN
13.     END

```

NOTE: IF BOTH KD AND AM DO NOT HAVE SAME SIGN AND  
KD IS NOT 0, THEN RADIANS WILL BE WRONG

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	00001 V	1	AM	R	SCALR	*00003 V	DUMMY	DMT0R	R	SPR0G	00000 P	
DMT0R	R	SCALR	00000 V	1	KD	I	SCALR	*00002 V	DUMMY					

## LOCAL VARIABLES (2 WORDS):

0000C DMT0R 00001 A

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

0000C DMT0R

## EXTERNAL SUBPROGRAMS REQUIRED:

SIT0R SSETUP2

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	DEC WORDS	HEX
CONSTANTS:	16	0001C
LOCAL VARIABLES:	2	00002
TEMPS:	2	00002
	3	00003
TOTAL PROGRAM:	23	00017

```

1.      SUBROUTINE DNAV(DLAT,KSN,DLON,KWE,RLAT,RLONG,KK)
2.      C   VERSION 3 FEB 1974, CORRECT KEYPUNCH ERROR
3.      C   SUBROUTINE DNAV, CONVERTS ANNOTATED DECIMAL DEGREES TO RADIANS
4.      C           IF KK=0
5.      C           CONVERTS RADIANS TO ANNOTATED DECIMAL DEGREES
6.      C           IF KK=-1
7.      C
8.      C   ANNOTATION IS N,S, W, OR E.
9.      C
10.     DEGRA = 1.745329E-2
11.     RADEG = 57.29578
12.     JN=1HN
13.     JS=1HS
14.     JE=1HE
15.     JW=1HW
16.     IF(KK) 100,50,50
17.     C   CONVERT DECIMAL DEGREES TO RADIANS
18.     50  RLAT=DLAT*DEGRA
19.     RLON=DLON*DEGRA
20.     IF(KSN=JS)75,70,75
21.     C   SOUTH LATITUDE
22.     70  RLAT=-RLAT
23.     75  IF(KWE=JW)85,80,85
24.     C   WEST LONGITUDE
25.     80  RLON=-RLON
26.     85  CONTINUE
27.     RETURN
28.     C   CONVERT RADIANS TO ANNOTATED DECIMAL DEGREES
29.     100 ALAT= ABS(RLAT)
30.     ALON= ABS(RLON)
31.     CLAT= ALAT*RADEG
32.     CLON = ALON*RADEG
33.     IF(RLAT)430,432,432
34.     430 KSN=JS
35.     GO TO 435
36.     432 KSN=JN
37.     435 IF(RLON)440,442,442
38.     440 KWE=JW
39.     GO TO 445
40.     442 KWE=JE
41.     445 RETURN
42.     END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
ABS	R	SPRNG	INTRIN		ALAT	R	SCALR	00008 V	1	ALONG	R	SCALR	00009 V	1
CEGRA	R	SCALR	00002 V	1	DLAT	R	SCALR	*0000B V	DUMMY	DLBN	R	SCALR	*0000D V	DUMMY
CNAV	SPRNG		00000 F		DNAV	R	SCALR	0000C V	1	IFKK	I	SCALR	00001 V	1
JE	I	SCALR	00006 V	1	JN	I	SCALR	00004 V	1	JS	I	SCALR	*00005 V	1
JM	I	SCALR	00007 V	1	JWE	I	SCALR	*0000E V	DUMMY	KK	I	SCALR	*00011 V	DUMMY
KSN	I	SCALR	*0000C V	DUMMY	KW	I	SCALR	*000CE V	DUMMY	RADEG	R	SCALR	00003 V	1
PLAT	R	SCALR	*0000F V	DUMMY	RLONG	I	SCALR	*00010 V	DUMMY					

[illegible]

## LOCAL VARIABLES (11 WORDS):

00005 JS	00004 JN	00003 RADEG	00002 DEGRA	00001 IFKK	00000 CNAV
0000A JWE	0000A JWE	00009 ALONG	00008 ALAT	00007 JN	00006 JE

## BLANK COMMON (C WORDS)

**ENTRY POINTS:**

COCOC CNAV

**INTRINSIC SUBPROGRAMS USED:**

**ABS**

EXTERNAL SUBPROGRAMS REQUIRED:

95ETLPA

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

```

      DEC      HEX
      WORDS   WORDS
-----
GENERATED CODE:      70      C0C46
CONSTANTS:           6      C0C06
LOCAL VARIABLES:    11      C0C0B
      TEMPS:         8      C0C08
-----
TOTAL PROGRAM:      95      C0C5F

```

```

1.      SUBROUTINE DREC(VN,VE,RLAT,RLONG,TDIF)
2.      CC
3.      C   THIS SUBROUTINE TAKES A POSITION (RLAT,RLONG)
4.      C   AND DR S USING VELBCITIES (VN,VE), AND TIME
5.      C   DIFFERENCE (TDIF IN HOURS) TO A NEW POSITION
6.      C   WHICH IS STORED IN (RLAT,RLONG)
7.      C   WRITTEN BY A. FOLINSBEE
8.      C   USES: REARH
9.      C   R = REARH(RLAT)
10.     C   SPECK = .5144444
11.     C   TSEC = TDIF*3600.
12.     C   RLAT = RLAT+(VN*SPECK*TSEC/R)
13.     C   BLAT = ABS(RLAT)
14.     C   RLONG = RLONG+VE*SPECK*TSEC/(R*COS(BLAT))
15.     C   RETURN
16.     C   END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
ABS	R	SPR0G	INTRIN	1	BLAT	R	SCALR	C00C4 V	1	CBS	R	SPR0G	INTRIN	1
DREC	R	SCALR	C00C0 V	1	DREC	R	SPR0G	C00C0 P	1	R	R	SCALR	00001 V	1
REARH	R	SPR0G	EXTERN	1	RLAT	R	SCALR	*C00C7 V	DUMMY	RL0NG	R	SCALR	*00008 V	DUMMY
SPECK	R	SCALR	C0002 V	1	TDIF	R	SCALR	*00C09 V	DUMMY	TSEC	R	SCALR	00003 V	1
VE	R	SCALR	*C0006 V	DUMMY	VN	R	SCALR	*00C05 V	DUMMY					

## LOCAL VARIABLES (5 WORDS):

C00C0 DREC 000C1 R

00002 SPEDK 000C3 TSEC 00004 BLAT

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

C00C0 DREC

## INTRINSIC SUBPROGRAMS USED:

ABS COS

## EXTERNAL SUBPROGRAMS REQUIRED:

REARH SC0S 9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
36	C0C24
2	C0C02
5	C0C05
7	C0C07
50	C0C32

GENERATED CODE:  
 CONSTANTS:  
 LOCAL VARIABLES:  
 TEMPS:  
 TOTAL PROGRAM:

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
ABS	R	SPR8G	INTRIN	1	BLAT	R	SCALR	00004 V	1	C8S	R	SPR8G	INTRIN	1
DREC	R	SCALR	00000 V	1	DREC	R	SPR8G	00000 P	1	R	R	SCALR	00001 V	1
REARH	R	SPR8G	EXTRN	1	RLAT	R	SCALR	*00007 V	DUMMY	RLONG	R	SCALR	*00008 V	DUMMY
SPECK	R	SCALR	00002 V	1	TDIF	R	SCALR	*00009 V	DUMMY	TSEC	R	SCALR	00003 V	1
VE	R	SCALR	*00006 V	DUMMY	VN	R	SCALR	*00005 V	DUMMY					

## LOCAL VARIABLES (5 WORDS):

00000 DREC 00001 R

00004 BLAT

00003 TSEC

00002 SPECK

BLANK COMMON (0 WORDS)

ENTRY POINTS:

00000 DREC

## INTRINSIC SUBPROGRAMS USED:

ABS

C8S

## EXTERNAL SUBPROGRAMS REQUIRED:

REARH 9C8S 9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	36	00024
CONSTANTS:	2	00002
LOCAL VARIABLES:	5	00005
TEMPS:	7	00007
TOTAL PROGRAM:	50	00032

```

1.      SUBROUTINE DY2M(ID,IY,M0,IDAY)
2.      C  DY2M TAKES CONSECUTIVE DAYS AND THE YEAR AND CHANGES THEM INTO DAYS & MONTHS
3.      DIMENSION MYDAY(13)
4.      DATA MYDAY/1,32,60,91,121,152,182,213,244,274,305,335,365/
5.      C  DETERMINE IF LEAP YEAR
6.      A=IY
7.      B=IY/4
8.      A=A/4.C
9.      IF(A=B) 12,10,12
10.     10  LEAP=1
11.         GO TO 13
12.     12  LEAP=0
13.     13  DO 14 I=3,13
14.         MYDAY(I)=MYDAY(I)+LEAP
15.     14  CONTINUE
16.         DO 15 I=1,12
17.         IF(MYDAY(I+1)-ID) 15,16,16
18.     16  M0=I
19.         IDAY=ID-MYDAY(I)+1
20.         GO TO 17
21.     15  CONTINUE
22.      C  RETURN M0=0 AND IDAY=0 IF ID GT 365+LEAP
23.      M0=0
24.      IDAY=0
25.     17  DO 18 I=3,13
26.         MYDAY(I)=MYDAY(I)-LEAP
27.     18  CONTINUE
28.      RETURN
29.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	0000E V	1	B	R	SCALR	0000F V	1	DY2M			00000 P	
DY2M	R	SCALR	00000 V	1	I	I	SCALR	00011 V	1	ID			00012 V	DUMMY
IDAY	I	SCALR	*00015 V	DUMMY	IY	I	SCALR	*00013 V	DUMMY	LEAP			00010 V	1
MO	I	SCALR	*00014 V	DUMMY	MYDAY	I	ARRAY	00001 V	13					

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
10	00014	12	00017	15	0002F
17	00037	18	0003D	16	00026

## LOCAL VARIABLES (18 WORDS):

CO00C DY2M	00001 MYDAY	0000E A	0000F B	00010 LEAP	00011 I
------------	-------------	---------	---------	------------	---------

BLANK COMMON (0 WORDS)

ENTRY POINTS:

CO00C DY2M

EXTERNAL SUBPROGRAMS REQUIRED:

SIT0R SSETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	DEC WORDS	HEX WORDS
CONSTANTS:	64	CO04C
LOCAL VARIABLES:	2	CO002
TEMP:	18	CO012
	5	CO005
TOTAL PROGRAM:	89	CO059

```

1. SUBROUTINE ENDLT(ICNT,CLAT0,CLAB0,CL0LE,CL0RI,IDL,ILI,IAR)
2. C      VERSION OF 10 MAY 72
3.      BLTPLY 'ENDLT OF 10 MAY 72 '
4.      BLTPLY CLAT0,CLAB0,CL0LE,CL0RI
5. C-----
6.      DIMENSION IA(20),IB(5)
7.      DATA ITERI,IBL/'EITP', '  '
8. C -----
9.      IIN=105
10.     ICNT=1
11.     RADEG = 57.29578
12. C -----
13.     IF(IAR.EQ.C)GOTO100
14. C
15.     CALL ARLIM(IIN,IIBUT,CLAT0,CLAB0,CL0LE,CL0RI)
16.     CLAT0 = CLAT0*RADEG
17.     CLAB0 = CLAB0*RADEG
18.     CL0LE = CL0LE*RADEG
19.     CL0RI = CL0RI*RADEG
20. C -----
21.     100 CONTINUE
22.     ALAT0P=CLAT0+90. ; LAT0P=ALAT0P
23.     ALAB0T=CLAB0+90. ; LAB0T=ALAB0T
24.     AL0LE=CL0LE+180. ; L0LE=AL0LE
25.     AL0RI=CL0RI+180. ; L0RI=AL0RI
26.     LA101=LAT0P/10 ; LA102=LAB0T/10
27.     L0101=L0LE/10 ; L0102=L0RI/10
28. C
29.     D0101I=LA101,LA102,-1
30.     D0102J=L0102,L0101,-1
31.     D0103K=9,0,-1
32.     D0104L=9,C,-1
33.     LAC=I*10+K ; L0C=J*10+L
34.     IF((LAC.LT.LAB0T).OR.(LAC.GT.LAT0P))GOTO104
35.     IF((L0C.LT.L0LE).OR.(L0C.GT.L0RI))GOTO104
36.     WRITE(ILI,105)LAC,L0C
37.     105 FORMAT(2I3)
38.     104 CONTINUE
39.     103 CONTINUE
40.     102 CONTINUE
41.     101 CONTINUE
42. C
43.     2 CONTINUE
44.     READ(IIN,1)IA
45.     1 FORMAT(20A4)
46. C -----
47.     IF(IA(1).EG.ITERI)WRITE(IDL,12)(IA(I),I=1,5) ; GOTO3
48.     IF(IA(1).EG.IBL)ICNT=ICNT+1
49.     * ; WRITE(IDL,12)(IA(I),I=1,5) ; GOTO2
50. C -----
51.     D010I=1,4
52.     IP=5*(I-1)+1
53.     IF(IA(IP).EG.IBL)GOTO10
54.     D011J=1,5
55.     IB(J)=IA(IP+J-1)
56.     11 CONTINUE
57.     WRITE(IDL,12)IB
58.     12 FORMAT(5A4)
59.     10 CONTINUE

```

60.           G0T02  
61.   C -----  
62.       3 REWIND IDL ; REWIND ILI ; RETURN  
63.       END

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
1	000AC	2	000A6	3	000F5	10	000F0	11	000E5
100	00049	101	000A2	102	000E	103	0009B	104	00091

LOCAL VARIABLES (EO WORDS):

0001C	ITERI	0001B	IBL	0001C	IIN
0001E	I18T	00021	AL48T	00022	LA80T
0002A	L81C2	00027	LA101	00028	LA102
0002B	L81C2	0002D	K	0002E	L
0003C	L8C	00031	IP		
0001A	IB	0001A	ITERI		
0001F	AL48P	0002C	LAT8P		
00025	AL6R1	00026	L8R1		
0002B	I	0002C	J		
00031	IP				

BLANK COMMON (C WORDS)

**ENTRY POINTS:**

CCCC ENCLT

EXTERNAL SUBPROGRAMS REQUIRED:

ARLIN	F:101	F:102	F:103	F:104	F:105	F:106	F:108
9RCREAD	9RCWRIT	9EAD10L	910DATA	918LUSA	SPRINT	9REIND	9RT01
9SETLFA							

HIGHEST ERROR SEVERITY: C (NO ERRORS)

```

DEC      250
WORDS   -----
      4
CONSTANTS:
LOCAL VARIABLES: 50
TERMS: 5
TOTAL PROGRAM: 313
      4
CCCF A
CCCO A
CCCO 2
CCCO 3
CCCO 9
CC139

```

```

1.      SUBROUTINE EVIL (JIBUT,I,IBAD,KDA,KM0,KYR,KHM)
2.      C  SUBROUTINE EVIL, CHECKS STATUS INDICATOR, WRITES ERROR MESSAGES,
3.      C      AND RETURNS INDICATOR FOR BAD READS(IBAD)
4.      C  IBAD=0, ALL OK
5.      CC
6.      CC
7.      IBAD=0
8.      GO TO (110,120,130,140,150,160),I
9.      12C  WRITE(JIBUT,122)KDA,KM0,KYR,KHM
10.     122  FORMAT('EOF FOUND',3I3,15)
11.     CS   PAUSE 122
12.      IBAD=2
13.      GO TO 110
14.     13C  WRITE(JIBUT,132)KDA,KM0,KYR,KHM
15.     132  FORMAT('EOT FOUND',3I3,15)
16.     CS   PAUSE 132
17.      IBAD=2
18.      GO TO 110
19.     14C  WRITE(JIBUT,142)KDA,KM0,KYR,KHM
20.     142  FORMAT('PARITY ER',3I3,15)
21.      IBAD=-1
22.      GO TO 110
23.     15C  WRITE(JIBUT,152)KDA,KM0,KYR,KHM
24.     152  FORMAT('FMT ER',3I3,15)
25.      IBAD=-1
26.      GO TO 110
27.     16C  WRITE(JIBUT,162)I,KDA,KM0,KYR,KHM
28.     162  FORMAT('ER I=',12,3I3,15)
29.      IBAD=-1
30.     11C  RETURN
31.      END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
EVIL	----	----	-----	-----	EVIL	----	----	-----	-----	I	----	----	-----	-----
IBAC	I	SCALR	*CCCC F	DUMMY	JBLT	R	SCALR	CCCC V	1	KDA	I	SCALR	*00002 V	DUMMY
KPY	I	SCALR	*CCCC V	DUMMY	KM8	I	SCALR	*CCCC V	DUMMY	KYR	I	SCALR	*00004 V	DUMMY
													*00006 V	DUMMY

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
110 CCC6C	-----	122 CC01F	-----	132 00030	-----
142 CCC41	120 CC017	152 CC052	130 CC028	162 00064	140 00039
	15C CC04A		160 CC05B		

## LOCAL VARIABLES (1 WORD):

CCCC EVIL

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

CCCC EVIL

## EXTERNAL SUBPROGRAMS REQUIRED:

F:102	F:104	F:106	F:108	9BCDWRT	910DATA	9SETUPN

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
109	CCCC6D
0	CCCCC
1	CCCC1
8	CCCC8
118	CCCC76

TOTAL PROGRAM: 118

```

1.      SUBROUTINE EXTD (CX, CXP, CY, CYP, BX, BY, IND)
2.      C
3.      C
4.      IND = 0
5.      D = ((CX - CXP)**2) + ((CY - CYP)**2)
6.      A = ((CX - BX)**2) + ((CY - BY)**2)
7.      B = ((CXP - BX)**2) + ((CYP - BY)**2)
8.      IF (A - D) 20, 20, 10
9.      10  IND = 1
10.     RETURN
11.     20  IF (B - D) 30, 30, 25
12.     25  IND = 1
13.     30  RETURN
14.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
A	R	SCALR	00002	V	B	R	SCALR	00003	V	6X	R	SCALR	00008	V
BY	R	SCALR	*00009	V	CX	R	SCALR	*00004	V	CXP	R	SCALR	*00005	V
CY	R	SCALR	*00006	V	CYP	R	SCALR	*00007	V	D	R	SCALR	00001	V
EXTC	R	SCALR	00000	V	EXTD	R	SPR0G	00000	P	IND	I	SCALR	*0000A	V

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
10 0002A	20 0002D	25 00030	30 00032		

## LOCAL VARIABLES (4 WORDS):

00000 EXTD	00001 D	00002 A	00003 B
------------	---------	---------	---------

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

00000 EXTD

## EXTERNAL SUBPROGRAMS REQUIRED:

SSETLFP

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	51	00033
CONSTANTS:	0	00000
LOCAL VARIABLES:	4	00004
TEMPS:	5	00009
TOTAL PROGRAM:	64	00040

```

1.      SUBROUTINE FIND(LIMDA,LIMMB,LIMYR,LIMHM,
2.      1  INDA,INMB,INYR,INHM,INDIC)
3.      C
4.      C  INDICATES WHETHER INPUT DATE < , = , > LIMIT DATE
5.      C  NO COMMON REQUIRED
6.      C  LIMDA,LIMMB,LIMYR,LIMHM ARE LIMIT DAY,MONTH, YEAR, TIME
7.      C  INDA,INMB,INYR,INHM ARE INPUT DAY,MONTH, YEAR, TIME
8.      C  INDIC IS INDICATOR
9.      C
10.     C  IF LIMYR = 99, NO COMPARISON IS MADE
11.     C  IF LIMMB = 99, ONLY YEARS ARE COMPARED
12.     C  IF LIMDA = 99, ONLY YEARS AND MONTHS ARE COMPARED
13.     C  IF LIMHM = 9999, DATES ARE COMPARED BUT TIMES ARE IGNORED
14.     C  NOTE: ALL FOUR LIMITS SHOULD BE GIVEN
15.     C
16.     C  INDICATOR SETTINGS:
17.     C  INDIC = -1 MEANS INPUT DATE IS BEFORE LIMIT DATE
18.     C  INDIC = 0 MEANS INPUT DATE IS EQUAL TO LIMIT DATE
19.     C  INDIC = +1 MEANS INPUT DATE IS AFTER LIMIT DATE
20.     C
21.     IF(LIMYR-99)100,300,100
22.     100 IF(INYR-LIMYR)199,101,201
23.     101 IF(LIMMB-99)102,200,102
24.     102 IF(INMB-LIMMB)199,103,201
25.     103 IF(LIMDA-99)104,200,104
26.     104 IF(INDA-LIMDA)199,105,201
27.     105 IF(LIMHM-9999)106,200,106
28.     106 IF(INHM-LIMHM)199,200,201
29.     199 INDIC = -1
30.     RETURN
31.     200 INDIC = 0
32.     RETURN
33.     201 INDIC = +1
34.     300 RETURN
35.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
FINC	----	----	----	----	FINC	----	----	----	----	INDA	----	----	----	----
INDIC	SPR8G	CC000 F	CC000	F	INPM	R	SCALR	00000 V	1	INMB	I	SCALR	*00005 V	DUMMY
INMR	I	SCALR	*00009 V	DUMMY	LIMDA	I	SCALR	*00008 V	DUMMY	LIMHM	I	SCALR	*00006 V	DUMMY
LIMPB	I	SCALR	*00007 V	DUMMY	LIMYR	I	SCALR	*00001 V	DUMMY					
	I	SCALR	*00002 V	DUMMY										

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
100 C000F	----	102 C0016	----	104 0001D	----
106 C0024	101 C0013	200 C002B	103 C001A	300 00030	00021
	199 C0028		201 C002E		

LOCAL VARIABLES (1 WORD):

CCCCC FIND

BLANK COMMON (0 WORDS)

ENTRY POINTS:

CCCCC FIND

EXTERNAL SUBPROGRAMS REQUIRED:

SSETCFN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
49	00031
0	00000
1	00001
10	0000A
60	0003C

GENERATED CODE:  
 CONSTANTS:  
 LOCAL VARIABLES:  
 TEMPS:  
 TOTAL PROGRAM:

```

1.      SUBROUTINE FLD2(KK,ITAPE,RLAT,RLONG,REG)
2.      C      VERSION 6 MAY 1974
3.      C      TO IMPLEMENT MODIFICATIONS BY FOLIMSBEE OF 24 APRIL 73
4.      C
5.      C      SUBROUTINE FLD2, CALCULATES REGIONAL FREE-AIR
6.      C      ANOMALIES FROM SPHERICAL HARMONIC COEFFICIENTS
7.      C      ENTERED AT RUN TIME
8.      C
9.      C
10.     DIMENSION S(25,25),C(25,25),BAR(25,25),F(25,25),SP(25)
11.     DIMENSION CP(25),FM(25),FN(25)
12.     C
13.     C      SSW(4) LP TO LIST INTERMEDIATE VALUES
14.     C      SSW(5) UP TO LIST BN,BV
15.     C
16.     C      TO CHANGE ORDER OF COEFFICIENTS REPLACE DIMENSION
17.     C      STATEMENTS BY ORDER + 1, AND SET NDIM=ORDER + 1
18.     C
19.     C      A BLANK CARD MUST FOLLOW COEFFS TO INDICATE THEIR COMPLETION
20.     C
21.     C
22.     C      KK=C FOR INITIAL ENTRY TO FLD2
23.     C      KK=1 HENCEFORTH
24.     C
25.     C      USES F4LIBS FLOAT, SQRT, SIN, COS, ATAN
26.     C
27.     IF(KK)500,999,500
28.     999  CONTINUE
29.     NDIM=25
30.     IIN = 105
31.     II0LT = 108
32.     REFG=980000.0
33.     OUTPUT 'FLD2'
34.     WRITE(II0LT,1)
35.     1  FORMAT('O N      M      C(N,M)  S(N,M)' )
36.     ISET=C
37.     MAXN=C
38.     22  DO 324 N=1,NDIM
39.     DO 324 M=1,N
40.     C(N,M)=0.
41.     S(N,M)=0.
42.     324  CONTINUE
43.     325  CONTINUE
44.     READ(ITAPE,302,END=30) N,M,CTEMP,STEMP
45.     IF(N) 30,30,25
46.     302  FORMAT (I2,2X,I2,2X,E11.4,2X,E11.4)
47.     25  C(N+1,M+1)=CTEMP
48.     S(N+1,M+1)=STEMP
49.     WRITE(II0LT,3) N,M,C(N+1,M+1),S(N+1,M+1)
50.     3  FORMAT(1X,I5,I5,2E11.4)
51.     55  IF(N-MAXN)325,325,26
52.     26  MAXN=N
53.     GO TO 325
54.     30  CONTINUE
55.     C 30  NDIM SHOULD BE SAME AS SIZE OF DIMENSIONED ARRAYS
56.     MAX1=MAXN+1
57.     CCCC ISET=C INDICATES FIRST CALL TO PROGRAM
58.     C      RENORMALIZE COEF IF THIS IS FIRST CALL
59.     IF (ISET) 500,2,500

```

```

60.      2  CONTINUE
61.      ISET=1
62.      DO 17 N=1,NDIM
63.      DO 17 M=2,NDIM
64.      P(N,M)=1.0
65.      17  CONTINUE
66.      BAR(1,1)=-1.0
67.      DO 20 N=2,NDIM
68.      FN(N)=N
69.      FM(N)=N-1
70.      BAR(N,1)=BAR(N-1,1)*FLOAT(2*N-3)/FLOAT(N-1)
71.      JT=2
72.      DO 20 M=2,N
73.      BAR(N,M)=BAR(N,M-1)*SGRT(FLOAT((N-M+1)*JT)/FLOAT(N+M-2))
74.      C
75.      C      BAR(N,M) ARE FACTORS TO RENORMALIZE CJS
76.      C
77.      JT=1
78.      20  CONTINUE
79.      DO 21 N=2,MAX1
80.      DO 21 M=1,N
81.      C(N,M)=C(N,M)*BAR(N,M)*SGRT(FLOAT(2*N-1))
82.      S(N,M)=S(N,M)*BAR(N,M)*SGRT(FLOAT(2*N-1))
83.      21  CONTINUE
84.      C
85.      C      NOW FINISHED WITH BAR, WILL USE LATER TO STORE CONST FOR
86.      C      RECURSION RELATION
87.      P(1,1)=1.
88.      SP(1)=0.
89.      CP(1)=1.
90.      RAD=.572957795E+02
91.      A=6378.388
92.      FLAT=1.0-1.0/C/297.
93.      A2=A**2
94.      A4=A**4
95.      B2=(A*FLAT)**2
96.      A2B2=A2*(1.-FLAT**2)
97.      A4B4=A4*(1.-FLAT**4)
98.      BAR(2,1)=0.
99.      BAR(2,2)=0.
100.     DO 24 N=3,NDIM
101.     DO 24 M=1,N
102.     BAR(N,M)=FLOAT((N-2)**2-(M-1)**2)/FLOAT((2*N-3)*(2*N-5))
103.     C
104.     24  CONTINUE
105.     RETURN
106.     C      WE HAVE NOW SET UP MOST CONSTANTS ARRAYS, ETC.
107.     C      THE PROGRAM COMES TO THIS POINT IF IT HAS BEEN ENTERED
108.     C      PREVIOUSLY
109.     C
110.     500  SINLA = SIN(RLAT)
111.     CP(2)=COS(RLONG)
112.     SP(2)=SIN(RLONG)
113.     DO 51 M=3,MAX1
114.     SP(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)
115.     CP(M)=CP(2)*CP(M-1)-SP(2)*SP(M-1)
116.     51  CONTINUE
117.     SNLA2=SINLA**2
118.     DEN2=A2-A2B2*SNLA2
119.     DEN=SGRT(DEN2)

```

```

120.      FAC=B2/A2
121.      THETA=ATAN(FAC*SINLA/(1.E-30+SGRT(1.-SNLA2)))
122.      R=SGRT((A4-A4B4*SNLA2)/DEN2)
123.      CT=SIN(THETA)
124.      ST = COS(THETA)
125.      ABR=1.0
126.      AR=ABR**2
127.      BV=0.
128.      DO 54 N=2,MAX1
129.      AR=ABR*AR
130.      DO 54 M=1,N
131.      IF(N-M) 112,111,112
132.      111 F(N,N)=ST*F(N-1,N-1)
133.      GO TO 113
134.      112 IF (N-2) 2011,2012,2011
135.      2012 GF=1.
136.      GO TO 2010
137.      2011 GF=F(N-2,M)
138.      2010 F(N,M)=CT*F(N-1,M)-BAR(N,M)*GF
139.      113 FNM=F(N,M)*AR
140.      TEMP=C(N,M)*CP(M)+S(N,M)*SP(M)
141.      BV=BV+TEMP*FLBAT(N-2)*FNM
142.      54 CONTINUE
143.      REG=-BV
144.      REG=REG*REFG
145.      IF(ISK(5)) 306,306,403
146.      403 WRITE(IIOUT,405)BV,COSD
147.      405 FORMAT('BV=',E11.4,2X,'COSD=',F6.3)
148.      306 RETURN
149.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	SCALR		00A36 V	1	AOR	R	SCALR	00A46 V	1	AR	R	SCALR	00A47 V	1
ATAN	R SPRBG		INTRIN	1	A2	R	SCALR	00A38 V	1	A2B2	R	SCALR	00A38 V	1
A4	R SCALR		00A39 V	1	A4B4	R	SCALR	00A3C V	1	BAR	R	SCALR	00A39 V	625
BV	R SCALR		00A48 V	1	B2	R	SCALR	00A3A V	1	C	R	ARRAY	00A43 V	625
CBS	R SPRBG		INTRIN	1	CBSC	R	SCALR	00A4C V	1	CP	R	ARRAY	00A4E V	25
CT	R SCALR		00A44 V	1	CTEMP	R	SCALR	00A31 V	1	DEN	R	SCALR	00A40 V	1
DEN2	R SCALR		00A3F V	1	FAC	R	SCALR	00A41 V	1	FLAT	R	SCALR	00A37 V	1
FLC2	R SCALR		00C00 V	1	FLC2	R	SPRBG	00C00 P	1	FLGAT	R	SPRBG	INTRIN	1
FM	R ARRAY		00A2A V	25	FN	R	ARRAY	00A10 V	25	GP	R	SCALR	00A49 V	1
IIN	I SCALR		00A2A V	1	ITOLT	I	SCALR	00A2B V	1	ISET	I	SCALR	00A42 V	1
ISW	I SPRBG		EXTERN	1	ITAPE	I	SCALR	00A4E V	DUMMY	JT	I	SCALR	00A34 V	1
KK	I SCALR		00A4D V	DUMMY	M	I	SCALR	00A30 V	1	MAXN	I	SCALR	00A4E V	1
MAX1	I SCALR		00A33 V	1	N	R	SCALR	00A4A V	1	NDIM	I	SCALR	00A29 V	1
P	R ARRAY		00A34 V	625	PNM	R	SCALR	00A4A V	1	REG	R	SCALR	00A43 V	1
RAD	R SCALR		00A35 V	1	REFG	R	SCALR	00A2C V	1	S	R	ARRAY	00001 V	DUMMY
RLAT	R SCALR		00A35 V	DUMMY	RLONG	R	SCALR	00A2F V	1	SINLA2	R	SCALR	00A3E V	625
SIN	R SPRBG		INTRIN	1	SINLA	R	SCALR	00A3D V	1	ST	R	SCALR	00A45 V	1
SP	R ARRAY		00C05 V	25	SRT	R	SPRBG	INTRIN	1	THETA	R	SCALR	00A42 V	1
STEMP	R SCALR		00A32 V	1	TEMP	R	SCALR	00A4B V	1					

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
1	0001C	3	0006D	17	00089	20	000BE
22	0002B	25	00053	26	00076	30	00079
54	001BB	111	0018A	112	00191	113	001A6
306	001DE	325	0003E	403	001CF	405	001D5
959	000CA	2011	00197	2012	00194	500	00137

## LOCAL VARIABLES (2637 WORDS):

0000C FLC2	00001 S	004E3 BAR	00754 P	009C5 SP
0000E CP	009F7 FM	00A28 NDIM	00A2A IIN	00A2B I1OUT
00A2C REFG	00A2D ISET	00A2E N	00A30 M	00A31 CTEMP
00A32 STEMP	00A33 MAX1	00A34 JT	00A36 A	00A37 FLAT
00A38 A2	00A39 A4	00A3A B2	00A3C A4B4	00A3D SINLA
00A3E SINLA2	00A3F DEN2	00A4C DEN	00A42 THETA	00A43 R
00A44 CT	00A45 ST	00A46 ABR	00A48 BV	00A49 GP
00A4A PNY	00A4E TEMP	00A47 AR		

## BLANK COMMON (C WORDS)

## ENTRY POINTS:

0000C FLC2

## INTRINSIC SUBPROGRAMS USED:

ATAN	CBS	FLGAT	SIN	SRT
------	-----	-------	-----	-----

## EXTERNAL SUBPROGRAMS REQUIRED:

ISH	F:101	F:102	F:103	F:104	F:105	F:106	F:108
9ATAN1	9BCDRDEE	9BCDWRIT	9CBS	9ENDI0L	9I0DATA	9IT0R	9PRINT
9PARRI	9SETUPN	9SIN	9SQRT				

## HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	479	C01DF
CONSTANTS:	8	C0C08
LOCAL VARIABLES:	2637	C0A4D
TEMPS:	9	C0C09
TOTAL PROGRAM:	3133	CCC3D

```

1.      SUBROUTINE GETC(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAW,RLAT,
2.      ,RLONG,KGDA,KGM0,KGYR,KGHM,IE0D)
3.      C
4.      C      SUBROUTINE GETC,F0R READING MAGNETICS AT CALCM F0RMAT
5.      CHANGED 17 JUNE 1971 BY CM W00DING T0 ANN0TATE 8BS MAG
6.      C
7.      DIMENSION PLT(7)
8.      IE0D=0
9.      II0UT=108
10.     14 CALL ENCI0
11.     15 READ(ITAPE,16)KGYR,KGDA,KGM0,KGHM,DLAT,DLONG,DATA,B,FLD,DIS,DIR,
12.     1SPD
13.     16 F0RMAT(11X,I2,1X,2I2,1X,F6.1,F7.3,F8.3,1X,2F6.0,F5.0,3X,F7.1,1X,
14.     1F3.0,F4.1)
15.     CALL STAT(I)
16.     CALL EVIL(II0UT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0)
17.     IF(IBAD)14,30,65
18.     65 IE0D=1
19.     RETURN
20.     30 RLAT=DLAT*(1.0/57.29578)
21.     RLONG=DLONG*(1.0/57.29578)
22.     PLT(1)=KGHM
23.     PLT(2)=B
24.     PLT(3)=FLD
25.     PLT(4)=DIS
26.     PLT(5)=DIR
27.     PLT(6)=SPD
28.     PLT(7)=DATA
29.     KGDA0=KGDA
30.     KGM00=KGM0
31.     KGYR0=KGYR
32.     KGHM0=KGHM
33.     C      SELECT P0INT T0 BE PLOTTED
34.     IF(NX)110,I20,110
35.     110 DATAX=PLT(NX)
36.     120 DATAY=PLT(NY)
37.     DATAZ=PLT(NZ)
38.     DATAW=PLT(NW)
39.     RETURN
40.     END

```

LABEL	HEX
-----	----
14 CCG17	
120 CCC71	
LABEL	HEX
-----	----
15 00C19	
16 CC029	
LABEL	HEX
-----	----
30 C00AD	
LABEL	HEX
-----	----
65 0004A	
LABEL	HEX
-----	----
110 0006E	
LABEL	HEX
-----	----
C006E	

CCCC	SETC	CCCC1	FLT	CCCCB	IIBLT	CCCCG	DLAT	0000A	DLNG	0000B	DATA
CCCC	B	CCCC	FLO	CCCC	DIR	CCCCF	SPO	00010	GPD	00011	I
CCCC12	IBAD	CCCC13	KGCAB	CCCC14	KGM69	CCCC15	KGY78	00016	KGH69		

COCCC G5YC

ENCIO	EVIL	STAT	F:101	F:103	F:105	9BCDREAD	910DATA
ENCIO	EVIL	STAT	F:101	F:103	F:105	9BCDREAD	910DATA

DEC	HEX
WORDS	WORDS
----	----
GENERATED CODE: 123	CC07B
CONSTANTS: 2	CC002
LOCAL VARIABLES: 23	CC017
TEPS: 17	CC011
----	----
TOTAL PROGRAM: 165	CC0A5

```

1.      SUBROUTINE GETF(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAW,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHM,IE0D)
3.      C
4.      C  SUBROUTINE GETF, FOR READING FIXES AT FIXSE FORMAT
5.      C
6.      C
7.      C  USES SUBROUTINES ENDI0(DUMMY), EVIL, DMT0R, STAT, ISW
8.      C  ISW AND STAT ASSUMED INITIALIZED IN MAIN PROGRAM
9.      C
10.     C
11.     C  DIMENSION PLT(8)
12.     C
13.     C  IE0D= 0
14.     C  II0LT = 108
15.     CS  II0LT=2
16.     CS  IF(ISW(15))13,14,13
17.     CS 13  PAUSE 15
18.     C 14  CALL ENDI0
19.     C 11  READ(ITAPE,16)KGDA,KGM0,KGYR,KGHM,ITDIF,LAT,
20.     C 1  RLATM,LONG,RL0M,K79,K80,KC1,KC2,KC3
21.     C 16  FORMAT(3I2,I4,1X,I3,1X,I3,1X,F6.2,1X,I4,1X,
22.     C 1  F6.2,1X,2I1,1X,3I1)
23.     C  CALL STAT(I)
24.     C  CALL EVIL(II0LT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0)
25.     C  IF (IBAD) 14, 30, 65
26.     C 65  IE0D = 1
27.     C  RETURN
28.     C 30  IF(ISW(12))55,60,55
29.     C 55  WRITE(II0LT,56)KGDA,KGM0,KGYR,KGHM
30.     C 56  FORMAT('DATE=',3I3,15)
31.     CS 60  IF(NLFYR)70,65,70
32.     C 60  CONTINUE
33.     C
34.     C
35.     C 70  RLAT=DMT0R(LAT,RLATM)
36.     C  RLONG=DMT0R(LONG,RL0M)
37.     C  PLT(4)=KGDA
38.     C  PLT(2)=KGM0
39.     C  PLT(3)=KGYR
40.     C  PLT(1)=KGHM
41.     C  PLT(5)=ITDIF
42.     C  PLT(6)=KC1
43.     C  PLT(7)=KC2
44.     C  PLT(8)=KC3
45.     C  KGDA0=KGDA
46.     C  KGM00=KGM0
47.     C  KGYR0=KGYR
48.     C  KGHM0=KGHM
49.     C  SELECTING DATA TO BE PLOTTED
50.     C  IF(NX)110,120,110
51.     C 110  DATAX= PLT(NX)
52.     C 120  DATAY= PLT(NY)
53.     C  DATAZ= PLT(NZ)
54.     C  DATAW= PLT(NW)
55.     C  RETURN
56.     C  END

```

```

DEC          WORDS
-----
GENERATED CODE:      151
CONSTANTS:           1
LOCAL VARIABLES:     26
TEPPS:               17
TOTAL PROGRAM:      195
-----
HEX          WORDS
-----
CC097
CC001
CC01A
CC011
CC0C3

```

```

1.      SUBROUTINE GETG(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAW,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHM,IE0D)
3.      C      VERSION 29 JULY 75, TO CORRECT FREAK INITIALIZATION PROBLEM
4.      C      THAT OCCURS ONLY WHEN READING FROM CARDS
5.      C      VERSION 19 SEPT 1974, FOR 67 GRAVITY FORMULA
6.      C      VERSION OF 7 AUGUST 1973, START CONVERSION TO 1967 G FORMULA
7.      C      GETG VERSION FEB 15 72
8.      C      ----- CHANGED INPUT FORMAT FOR BACKWARD CODE MANAGE PROBLEM
9.      C      ----- MODIF FOR NEW ABSTG FORMAT HANDLING
10.     C      GETG VERSION DEC 30 , 71
11.     C      GETG VERSION NOV 19 , 1971
12.     C      INPUT . NEW GSUM FORMAT .
13.     C      + 3 KEYS : LTKEY , LGKEY , IAKEY
14.     C
15.     C      MODIF ON NOV 12, 1971 BY MENGET J.M. TO INCLUDE :
16.     C      - USE OF DATA LOCATION TABLE
17.     C      - USE OF LABEL TAPES
18.     C      MOD AUG 16 1971 BY FOLINSBEE TO CORRECT ERROR ON CALLING ARGS OF GBLKI
19.     C      VERSION OF JUNE 29, 71 DOES NOT WRITE EOF ONTO OUTPUT DEVICE
20.     C      MODIFIED JUNE 28 TO READ( OR WRITE) BLOCKED DATA BY A FOLINSBEE
21.     C      VERSION OF APRIL 16 TO OPTIONALLY SUPPRESS REWIND OF ITAPE AND JTAPE GIN00040
22.     C      GIN00050
23.     C      WITH COMPLETE ARGUMENT LIST, ISBRC MOVED TO COME GIN00060
24.     C      AFTER DATE GIN00070
25.     C      GIN00080
26.     C      SUBROUTINE GETG, FOR GSUM FORMATTED DATA
27.     C      GIN00100
28.     C      GIN00110
29.     C      VERSION WITH DESIGNATION OF INPUT MAGNETIC TAPES
30.     C      BY USE OF SUBROUTINE MOUNT GIN00130
31.     C      GIN00140
32.     C      GIN00150
33.     C      SSW(12) UP TO LIST DATE IDENTIFICATION
34.     C      SSW(27) UP TO SUPPRESS REWIND OF TAPES AT START OF JOB
35.     C      * 1 SUPPRESS REWIND OF ITAPE GIN00240
36.     C      SSW(29) = 1 - TO READ AND TEST FOR SELECTED SOURCE CODE
37.     C      NUMBERS TO BE PROCESSED
38.     C      * 2 - TO READ AND TEST FOR SELECTED SOURCE CODE
39.     C      NUMBERS TO BE SKIPPED
40.     C
41.     C
42.     C      SSW(30) UP FOR INPUT DATA ON CARDS GIN00190
43.     C      SSW(40) UP TO PROCESS WITH BOUNDS USING DLT
44.     C      * 0 - PROCESS WITHOUT BOUNDS
45.     C      * 1 - PROCESS WITH BOUNDS USING THE DATA LOCATION TABLE
46.     C      SSW(60) UP TO PROCESS ONLY DATA WITH IFFC=4, ABSTRACTER OUTPUT GIN00210
47.     C      SSW(61) UP TO REPLACE FA,BG,ELEV,LAT, LONG WITH AVERAGED VALUES
48.     C      GIN00270
49.     C      USES ENDI8(DUMMY), EVIL, STAT, ISW GIN00280
50.     C      ASSUME ISW AND STAT INITIALIZED IN MAIN PROGRAM GIN00290
51.     C      GIN00300
52.     C      GIN00310
53.     C      GIN00320
54.     C      GIN00330
55.     C      DIMENSION IDTIN(20),IBK(20),ITK(20),IDENS(20),IDESC(17,20)
56.     C      DIMENSION IDTBT(20),JBK(20),JTK(20),JDENS(20),JDESC(17,20)
57.     C      EGLVANCE (DLAMI,DLAB0),(DLAMA,DLAT0),(DL0MI,DL0LE),
58.     C      * (DL0MA,DL0RI)
59.     C      DIMENSION PLT(20),IA(35),IAFMT(9),IASH(35),ISRC(16)
60.     C      DATA ITERI/'EITP'/
61.     C      DATA IFLAG/C/

```

```

60. C
61.     IF (IFLAG.NE.C) GOTO 50
62.     IFLAG=1
63. C
64. C
65. C ----- GETG INITIALISATION LOGIC
66. C
67.     IIN=105
68.     IIOUT = 108
69.     IEOD=0
70.     IDISC = 100
71.     KI = 1
72.     K8 = -2
73.     NEF = 1
74.     IFILE = 1
75. C
76.     DEGRA = 1.745329E-2
77.     RADEG=57.29578
78. C
79.     OUTPUT ' GETG VERSION 29 JULY 75 FOR 67 G FORMULA'
80.     IF (ISW(40).NE.C) CALL ENTCLT(J,DLAT0,DLAB0,DL0LE,DL0RI)
81.     * ; CALL SETSKP(INDICA) ; IDLT=0
82.     * ; NEF=1 ; IFILE=J-1 ; OUTPUT NEF,IFILE
83.     NZERO=C
84.     KGDA0=NZERO
85.     KGM00=NZERO
86.     KGYR0=NZERO
87.     KGHM0=NZERO
88. C     NRECT = N0. 0F RECORDS NOW WRITTEN ON PRESENT OUTPUT TAPE
89. C     NEF = N0. 0F FILE NOW BEING PROCESSED
90.     NRECT = NZERO
91.     IREC1 = 1
92. C
93.     IF (ISW(40).NE.C) GOTO 141C
94.     IF (ISW(30)) 404,404,1410
95.     404 J=1
96.     405 READ (IIN,406) IDTIN(J),IBK(J),ITK(J),IDENS(J),
97.     1 (IDESC(K,J),K=1,17)
98.     406 FORMAT(A4,1X,A1,1X,I1,1X,I3,17A4)
99.     IF (IDTIN(J).NE.ITERI) J=J+1; GO TO 405
100.     NEF=1
101.     IFILE=J = 1
102.     OUTPUT NEF,IFILE
103. C
104. C ----- CHECK SSW(29) TO SEE IF SOURCE CODE NUMBERS
105. C ----- ARE TO BE READ FOR DATA SELECTION
106. C
107.     IF (ISW(29).EG.C) GOTO 10
108.     READ(IIN,900) ISRC
109.     900 FORMAT(16I5)
110.     IF (ISW(29).EG.1) WRITE(IIOUT,912) ISRC ; GOTO 1410
111.     WRITE(IIOUT,913) ISRC
112.     913 FORMAT(1H0,10X,'SKIPPED SOURCE CODES = ',16I5)
113.     912 FORMAT(1H0,10X,'SELECTED SOURCE CODES = ',16I5)
114. C
115. C -----
116. C
117.     141C IF (ISW(30)) 10,10,2414
118.     1C IF (ISW(40).EG.C) GOTO 810
119.     READ(IDISC,406) IDTIN(1),IBK(1),ITK(1),IDENS(1),

```

GIN00390

GIN00410

GIN00430

GIN00470

GIN00480

GIN00490

GIN00500

GIN00510

GIN00520

GIN00530

GIN00540

GIN00550

GIN00560

GIN00580

GIN00590

GIN00600

GIN00610

GIN00620

GIN00630

GIN00640

GIN00650

```

120.      * (IDESC(K,1),K=1,17)
121. 810 IF (IDTIN(1).EG.ITER1) IE8D=1 ; RETURN
122.      CALL MOUNT(ITAPE,IDTIN(1))
123.      OUTPUT 'INPUT TAPE MOUNTED '
124.      WRITE (II8UT,1413) IDTIN(1),IBK(1),ITK(1),IDENS(1),
125.      1 (IDESC(K,1),K=1,17)
126. 1413 FORMAT (1X,A4,1X,A1,1X,I1,1X,I3,17A4)
127.      OUTPUT '-----'
128.      IF (ISW(27).EG.1) G8T82414
129.      REWIND ITAPE
130. 2414 CONTINUE
131. C
132. C ----- GETG INPUT LOGIC
133. C
134. 50 CALL ENDI8
135.      IF (ISW(30).EG.1) G8 T8 100
136.      IF (ISW(40).NE.0) G8 T8 700
137. 52 READ (ITAPE,11) IREC1,IS8RC,KGDA,KGM8,KGYR,KGHM,
138. 1 DLAT,DL8NG,ELEV,K977,8BSG,IDEF,FA,BG,TC,IELC,IGC,
139. 2 RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
140.      G8 T8 101
141. 100 CONTINUE
142.      READ (IIIN,469) IREC1,IS8RC,KGDA,KGM8,KGYR,KGHM,
143. 1 DLAT,DL8NG,ELEV,K977,8BSG,IDEF,FA,BG,TC,IELC,IGC,
144. 2 RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
145. 101 CONTINUE
146.      CALL STAT(I)
147. 710 CONTINUE
148.      CALL EVIL (II8UT,I,IBAD,KGDA8,KGM88,KGYR8,KGHM8)
149.      IF (IEAD) 50, 53, 575
150. 53 IF (IREC1-2) 600,70,600
151. 600 IF (IREC1-1) 50,610,50
152. C
153. C      CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF
154. C      THE 1967 INTERNATIONAL GRAVITY FORMULA
155. C
156. 610 CONTINUE
157.      RLAT = DLAT*DEGRA
158.      CG=3.2-(13.6*(SIN(ABS(RLAT))*2))
159. C      TO AVOID PLOTTING INVALID VALUES
160.      IF (FA.EG.999.AND.NX.EG.6) G8 T8 50
161.      FA=FA+CG
162.      IF (BG.EG.999.AND.NX.EG.7) G8 T8 50
163.      BG=BG+CG
164.      G8 T8 70
165. 575 IF (NEF = IFILE) 576, 577, 577
166. 576 NEF = NEF + 1
167.      IF (ISW(40).EG.C) G8T8820
168.      READ (IDISC,406) IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
169.      * (IDESC(K,NEF),K=1,17)
170. 820 CALL MOUNT(ITAPE,IDTIN(NEF))
171.      OUTPUT 'INPUT TAPE MOUNTED'
172.      WRITE (II8UT,1413) IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
173.      1 (IDESC(K,NEF),K=1,17)
174.      OUTPUT '-----'
175.      REWIND ITAPE
176.      G8 T8 50
177. C      END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
178. 577 IE8D=1 ; RETURN
179. 70 IF (ISW(12)) 73,73,71

```

GIN80770

GIN80780

GIN80790

GIN80820

GIN80830

GIN80950

GIN80960

GIN81020

GIN81030

GIN81090

GIN81100

GIN81110

GIN81120

GIN81240

GIN81250

GIN81270

GIN81280

GIN81290

GIN81300

GIN81310

GIN81390

```

180.      71  WRITE(IIOU,72)KGDA,KGM8,KGYR,KGHM
181.      72  FORMAT('DATE=',3I3,15)
182.      C
183.      C -----
184.      C
185.      73  IF(ISW(29).EG.C)G0T0173C
186.          IF(ISW(29).EG.2)G0T0170C
187.      C
188.      C -----      PROCESS ONLY SELECTED SOURCE CODES
189.      C
190.          D01650=1,16
191.          IF(ISRC(J).EG.C)G0T050
192.          IF(IS0RC=ISRC(J))1650,173C,1650
193.      165C  CONTINUE
194.          G0T050
195.      C
196.      C -----      IGNORE SELECTED SOURCE CODES
197.      C
198.      170C  D01710=1,16
199.          IF(ISRC(J).EG.C)G0T0173C
200.          IF(IS0RC=ISRC(J))1710,5C,1710
201.      171C  CONTINUE
202.      C
203.      C -----
204.      C
205.      173C  RLAT=DLAT*DEGRA
206.          RL0NG = DL0NG*DEGRA
207.          DEPTH = IDEP
208.          IF(IDEP)78,74,78
209.      74  HEIGT=ELEV
210.          G0 T0 80
211.      78  HEIGT = -DEPTH
212.      8C  CONTINUE
213.      C  CONVERTING FROM THE POTSDAM REF. NO. TO THE IGSN-71 REF. SYSTEM
214.      C  NX = 11 TO PLOT OBSERVED GRAVITY
215.          IF(NX.NE.11) G0 T0 81
216.          IF(IREC1.EG.2) G0 T0 81
217.          CALL 0BG (K977,0BSG,G0BS,K1)
218.          G0BS=G0BS-14.0
219.      81  CONTINUE
220.          BGC0M=BG+TC
221.          PLT(1)=KGMH
222.          PLT(2)=IS0RC
223.          PLT(3)=ELEV
224.          PLT(4)=DEPTH
225.          PLT(5)=HEIGT
226.          PLT(6)=FA
227.          PLT(7)=BG
228.          PLT(8)=TC
229.          PLT(9)=BGC0M
230.          PLT(10)=RFA
231.          PLT(11)=G0BS
232.          PLT(12)=HEIGT/BG
233.          IF(ISW(60))418,418,80C
234.      80C  IF(IFFC=10)50,801,50
235.      801  CONTINUE
236.          IF(ISW(61))109,109,105
237.      C  SET FA, BG, AND ELEV = AVERAGED VALUES FROM ABSTRACTER OUTPUT
238.      C  SET LAT AND LONG TO VALUES AT CENTER OF GRID AREA
239.      105  CONTINUE

```

GIN01400  
GIN01410

GIN01460

GIN01480  
GIN01490  
GIN01500

```

240.      DO 802 JK=1,35
241.      802 IASH(JK)=ISL(IA(JK),-24)
242.      CALL PKBY(IASH,IAFMT,35)
243.      DEC0DE(35,803,IAFMT)NINT,I,J,AVHE,AVFA,AVB0,N0BS,ITM,DIST
244.      803 F0RMAT(3I2,F8.1,2F6.1,I4,A1,F4.1)
245.  C
246.      AGRI=1./FL0AT(NINT) ; HGRI=AGRI/2
247.      IAX=LTKEY=89 ; IB=LGKEY=180
248.      ALV=FL0AT(IAX) ; 0LV=FL0AT(IB)
249.      CLAT=ALV+HGRI-FL0AT(I-1)*AGRI
250.      CL0NG=0LV+HGRI+FL0AT(J-1)*AGRI
251.      RLAT=CLAT*DEGRA
252.      RL0NG=CL0NG*DEGRA
253.      FLT(13)=AVHE ; FLT(14)=AVFA
254.      FLT(15)=AVB0 ; FLT(16)=N0BS
255.      FLT(17)=DIST
256.      109 CONTINUE
257.      418 CONTINUE
258.      KGDA0=KGDA
259.      KGM00=KGM0
260.      KGYR0=KGYR
261.      KGHM0=KGHM
262.      IF(NX)1100,1200,1100
263.      1100 DATA=FLT(NX)
264.      1200 DATA=FLT(NY)
265.      DATAZ=FLT(NZ)
266.      DATAW=FLT(NW)
267.  X      0UTPUT DATA
268.      RETURN
269.      11 F0RMAT(I1,I4,3I2,I4,2F9.4,F7.2,I3,F6.2,I5,2F6.1,F4.1,
270.      1 2I2,F6.1,I1,I2,35A1,1X,I1,2I3,I2)
271.      469 F0RMAT(I1,I4,3I2,I4,2F9.4,F7.2,I3,F6.2,I5,2F6.1,F4.1,
272.      1 2I2,F6.1/10X,I1,I2,35A1,1X,I1,2I3,I2)
273.  C
274.  C ----- GETG DLT INPUT LOGIC
275.  C
276.      700 CONTINUE
277.      IF(ICLT.EG.1)G0T0821
278.      READ(ICISC,812)NMAX ; IPRE = NMAX ; NPRE=1
279.      0UTPUT NMAX
280.      812 F0RMAT(I6)
281.      322 CONTINUE
282.      READ(ICISC,321,END=331)NBL0,ILAST,LAstr,0LMAX,0LMIN
283.      0UTPUT NBL0,ILAST,LAstr,0LMAX,0LMIN
284.      321 F0RMAT(4X,I6,I6,I3,2F9.4)
285.      IF(NBL0.EG.0)IDLT=0 ; G0T0575
286.      ALMIN=FL0AT(LAstr=90)
287.      ALMAX=ALMIN+1.
288.      0LARG=0LMAX-0LMIN
289.      CL0LP=CL0MA+0LARG
290.      CL00=CL0MI-0LARG
291.      IF((CL0LP.GE.0LMAX).AND.(CL00.LE.0LMIN))G0T0323
292.      IPRE=ILAST
293.      G0T0322
294.      323 CL0LP=CL0MA+1.
295.      CL00=CL0MI-1.
296.      IF((CL0LP.GE.ALMAX).AND.(CL00.LE.ALMIN))G0T0324
297.      IF(ALMAX.LT.CL00)G0T0343
298.      IPRE=ILAST
299.      G0T0322

```

GIN01510  
GIN01520  
GIN01530

GIN01640  
GIN01650  
GIN01660  
GIN01670  
GIN01680  
GIN01690

```

300.      324 IF(IPRE.EG.NMAX)GOTO325
301.      IBEG=IPRE+1
302.      GOTO326
303.      325 IBEG=1
304.      C ----- NINF = INDEX FIRST REC. TO READ
305.      C ----- NSUP = INDEX LAST REC. TO READ
306.      326 NFIR = NMAX*(NBL0-1)
307.      NINF=NFIR+IBEG
308.      NSUP=NFIR + ILAST
309.      C ----- AVOID TRYING TO READ REC. WHICH ARE ALREADY PROCESSED
310.      IF(NINF.LT.NPRE)NINF=NPRE
311.      NSKIP=NINF-NPRE
312.      CALL SKPREC(ITAPE,NSKIP,'FWD')
313.      GOTO(330,330,331,332,333)INDICA
314.      330 CONTINUE
315.      NPRE=NSUP+1
316.      ICNREC = NSUP-NINF +1 ; ICNT = 0
317.      821 IF(ICNT.LT.ICNREC)IDLT=1 ; GOTO870
318.      IDLT=0 ; GOTO322
319.      870 READ(ITAPE,11)IREC1,IS0RC,KGDA,KGM0,KGYR,KGHM,
320.      1 DLAT,DLONG,ELEV,K977,0BSG,IDEF,FA,BG,TC,IELC,IGC,
321.      2 RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
322.      ICNT = ICNT+1
323.      GOTO101
324.      343 WRITE(IIOUT,344)
325.      344 FORMAT(1H0,'TEST AREA ALREADY PROCESSED',/)
326.      IE0D=1 ; RETURN
327.      331 WRITE(IIOUT,345)
328.      345 FORMAT(1H0,'INCORRECT DLT TABLE - FOUND EOF WHILE PROCESSING
329.      * RECORDS')
330.      IE0D =1 ; RETURN
331.      333 WRITE(IIOUT,346)
332.      346 FORMAT(1H0,'INCORRECT DLT TABLE - FOUND END OF TAPE WHILE
333.      * SKIPPING RECORDS',/)
334.      IE0D=1 ; RETURN
335.      332 WRITE(IIOUT,347)
336.      347 FORMAT(1H0,'ERROR CONDITION WHILE SKIPPING RECORDS',/)
337.      IE0D=1 ; RETURN
338.      END

```

GIN02060

LABEL	HEX L8C	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
10	00CE	0031	50	0045	52	0051	53	00A1	70	0022
71	0027	002F	73	0034	74	0036	78	0065	80	0026
81	0027	0072	101	0032	105	003A	109	00FA	321	0079
322	0035	005D	324	003F	325	003B	326	00BB	330	00D3

331	CC41F	332	CC456	333	CC439	343	CC4CD	344	CC411	345	CC423
346	CC43C	347	CC45A	404	CC076	405	CC078	406	CC093	418	CC02FA
469	CC326	575	CC1C4	576	CC1C7	577	CC21F	600	CC01A4	610	CC01A7
700	CC33C	710	CC195	800	CC29A	801	CC29D	802	CC02A5	803	CC02BD
810	CC1C6	812	CC351	820	CC29A	821	CC3E2	870	CC03EB	900	CC00BE
912	CC0CE	913	CC0C3	1100	CC3C4	1200	CC3C7	1410	CC00EA	1413	CC012C
1650	CC246	1700	CC024A	1710	CC253	1730	CC257	2414	CC0145		

## LOCAL VARIABLES (1046 WORDS):

CC00C GETG	CC001 IDTIN	CC029 ITK	0003D IDENS	00051 IDESC
CC015 IBK	CC018 YBK	CC01E1 JDENS	001F5 JDESC	00349 DLAMI
CC015 JTK	CC034A DLAMA	CC34B DLAMI	0034B DLBLE	0034C DLORI
CC0361 IA	CC03C1 IFLAG	CC384 IAFMT	0038D IASH	00380 ISRC
CC03C2 IIN	CC03C7 NEF	CC3C3 I1OUT	003C4 IDISC	003C5 KI
CC03CE IFILE	CC03CD IDLT	CC3CS DEGRA	003CA RADEG	CC3CB J
CC03CE NZERO	CC03C3 ARECT	CC3CF KGDA0	003D0 KGM90	CC3D1 KGYR0
CC03CA K977	CC03C4 IRECT	CC3D5 K	003D6 ISBRC	CC3D7 DLAT
CC03EC IELC	CC03E1 IGC	CC3D8 0BSG	003DC IDEP	CC3DD FA
CC03E6 LTKEY	CC03E7 LGKEY	CC3E1 IGC	003E2 RFA	003E3 IREGC
CC03EC DEPTH	CC03E8 IKEY	CC3E7 LGKEY	003E8 IKEY	003E9 I
CC03F2 AVHE	CC03F3 AVFA	CC3EE G8BS	003EE G8BS	CC3EF BGC0M
CC03F7 DIST	CC03F8 AGRI	CC3F3 AVFA	003F4 AVB0	CC3F5 N8BS
CC03FC BLV	CC03FE CLAT	CC3F5 HGRI	003FA IAX	CC3FB I0
CC04C3 NBL0	CC04C4 ILAST	CC3FF CLONG	00400 NMAX	CC401 IPRE
CC04C9 ALMAX	CC04CF IBEG	CC4C5 LSTR	00406 0LMAX	CC407 0L4IN
CC04CE CLAB0	CC041C NFIR	CC4CB DL0UP	0040C DL0UP	CC40D DL0UP
CC0414 ICNREC	CC415 IC0NT	CC411 N1NF	00412 NSLP	CC413 NSKIP

## BLANK COMMON (C WORDS)

## ENTRY POINTS:

CCCC GETG

## INTRINSIC SUBPROGRAMS USED:

ABS	FL0AT	ISL	SIN
-----	-------	-----	-----

## EXTERNAL SUBPROGRAMS REQUIRED:

ENDI0	ENTCLT	EVIL	ISW	M0UNT	0BSG	PKBY	SETSKP
SXPREC	STAT	F:1C1	F:102	F:103	F:104	F:105	F:106
F:1C0	SECCRDCE	9BCDREAD	9BCCARIT	9CEC0DE	9ENDI0L	9IBDATA	9IALUSA
9IT0R	SPRINT	9REWIND	9SETLPA	9SIN			

NUMBER OF X CARDS IGNORED: 1

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	113C	C046A
CONSTANTS:	19	C0C13
LOCAL VARIABLES:	1C46	C0416
TEMPS:	18	C0C12
	-----	-----
TOTAL PROGRAM:	2213	C08A5

```

1.      SUBROUTINE GETGA(ITAPE,NX,DATA,NY,DATAY,NZ,DATAZ,
2.      *      RTOP,RRBT,RLFT,RRGT,RLAT,RLONG,IE80)
3.      DIMENSION IBUFIN(1184)
4.      DIMENSION KAVFA(36),KAVEL(36),KPTS(36)
5.      DIMENSION KCENEL(36),KCENFA(36)
6.      DIMENSION KCENLAT(36),KCENLONG(36)
7.      DATA INIT/0000/
8.      IF(INIT.NE.0) GO TO 200
9.      C      INITIALIZATION
10.     OUTPUT 'SUBROUTINE GETGA VERSION OF 25 OCT 75'
11.     INIT =1
12.     NCARD=105
13.     NIN=37
14.     IE80=0
15.     RADEG=57.2458
16.     DEGRA=1.7453E-2
17.     KDTOP=RTOP*RADEG+200
18.     KDBBT=RRBT*RADEG+200
19.     KDLFT=RLFT*RADEG+200
20.     KDRGT=RRGT*RADEG+200
21.     200 CONTINUE
22.     IF(NIN.LT.37) GO TO 250
23.     CALL BUFFER IN(ITAPE,0,IBUFIN(1),296,IKEY,NI)
24.     210 CONTINUE
25.     GO TO(211,213,214) IKEY
26.     211 OUTPUT 'WAITING' ; GO TO 210
27.     213 OUTPUT 'END ON ITAPE' ; GO TO 900
28.     214 OUTPUT 'ERROR ON INPUT' ; STOP
29.     215 CONTINUE
30.     DECODE(1184,1001,IBUFIN(1),ND)
31.     *      KLAT,KLONG,KDUM1,KDUM2,KDUM3,KDUM4,KDUM5,
32.     *      (KCENEL(1),KCENFA(1),KCENLAT(1),KCENLONG(1),
33.     *      KAVEL(1),KAVFA(1),KPTS(1),I=1,36)
34.     NIN=0
35.     IF((KLAT.GT.KDTOP).OR.(KLAT.LT.KDBBT)) NIN=37; GO TO 200
36.     IF((KLONG.LT.KDLFT).OR.(KLONG.GT.KDRGT)) NIN=37 ; GO TO 200
37.     250 CONTINUE
38.     GO TO (310,320,330,340,350) (NX=3)
39.     C      AVERAGE FREE AIR
40.     310 CONTINUE
41.     NIN=NIN+1
42.     IF(NIN.EQ.37) GO TO 200
43.     IF(KPTS(NIN).EQ.0) GO TO 310
44.     DATA=FLBAT(KAVFA(NIN)/10)
45.     GO TO 400
46.     C      AVERAGE ELEVATION
47.     320 CONTINUE
48.     NIN=NIN+1
49.     IF(NIN.EQ.37) GO TO 200
50.     IF(KPTS(NIN).EQ.0) GO TO 320
51.     DATA=FLBAT(KAVEL(NIN))
52.     GO TO 400
53.     C      CENTRAL FREE AIR
54.     330 CONTINUE
55.     NIN=NIN+1
56.     IF(NIN.EQ.37) GO TO 200
57.     IF(KPTS(NIN).EQ.0) GO TO 330
58.     DATA=FLBAT(KCENFA(NIN)/10)
59.     GO TO 400

```

```

60.      C          CENTRAL ELEVATION
61.      340 CONTINUE
62.      NIN=NIN+1
63.      IF(NIN.EQ.37) GO TO 200
64.      IF(KPTS(NIN).EQ.0) GO TO 340
65.      DATA=FLBAT(KCENEL(NIN))
66.      GO TO 400
67.      C          NUMBER OF POINTS
68.      350 CONTINUE
69.      NIN=NIN+1
70.      IF(NIN.EQ.37) GO TO 200
71.      IF(KPTS(NIN).EQ.0) GO TO 350
72.      DATA=FLBAT(KPTS(NIN))
73.      400 CONTINUE
74.      RLAT=(KLAT+FLBAT(KCENLAT(NIN))/100.-200)*DEGRA
75.      RLONG=(KLONG+FLBAT(KCENLONG(NIN))/100.-200)*DEGRA
76.      RETURN
77.      C          END OF FILE
78.      900 CONTINUE
79.      IE9D=1
80.      RETURN
81.      1001 FORMAT(37(I6,I6,2I2,2I6,I4))
82.      END

```

LABEL	HEX L8C
----	-----
200	00042
250	000B1
400	000FA
LABEL	HEX L9C
----	-----
210	0004D
310	000BD
900	00111
LABEL	HEX L8C
----	-----
211	00056
320	000CA
1001	00114
LABEL	HEX L9C
----	-----
213	0035F
330	000D6
LABEL	HEX L8C
----	-----
214	00069
340	000E3
LABEL	HEX L9C
----	-----
215	00074
350	000EF

LOCAL VARIABLES (1457 WORDS):

00000	GETGA	00001	TRUFIN	004A1	KAVFA	004C5	KAVEL	004E9	KPTS	0050D	KCEMEL
00531	KCENFA	00555	KCENLAT	00579	KCENLANG	0059D	INIT	0059E	NCARD	0059F	NIN
005A0	RADEG	005A1	DEGRA	005A2	KDT8P	005A3	KDB8T	005A4	KDLFT	005A5	KDGST
005A6	IKEY	005A7	NI	005A8	ND	005A9	KLAT	005AA	KLONG	005AB	KDJM1
005AC	KDJM2	005AD	KDJM3	005AE	KDJM4	005AF	KDJM5	005B0	I	005B1	

BUFFERIN  
 9SETUPN  
 F:108  
 9STEP  
 9DECADE

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	284	0011C
CONSTANTS:	8	00008
LOCAL VARIABLES:	1457	005B1
TEMPS:	16	00010
	-----	-----
TOTAL PROGRAM:	1765	006E5

```

1.      SUBROUTINE GETGC(ITAPE,NX,DATAX,NY,DATAY,NZ,DATAZ,
2.      *      RTOP,RTOT,RLFT,RRGT,RLAT,RLONG,IE9D)
3.      C      VERSION OF 10 JAN 76 TO DO NX PROPERLY
4.      C      AND TO IMPLEMENT SSW 46
5.      C      ORIGINAL VERSION 13 SEPT 75
6.      DIMENSION IBUFIN(6,300)
7.      DIMENSION KODE(300),KDLAT(300),KDLONG(300),KELDEP(300),KFA(30)
8.      DIMENSION KAVFA(10,10),KAVEL(10,10),NPTS(10,10)
9.      DIMENSION H0WFAR(10,10),KCENLAT(10,10),KCENLONG(10,10)
10.     DIMENSION KCENEL(10,10),KCENFA(10,10)
11.     DIMENSION NCENEL(100),NCENFA(100),NCENLAT(100),NCENLONG(100)
12.     DIMENSION NAVFA(100),NAVEL(100)
13.     EQUIVALENCE (KAVFA,NAVFA)
14.     EQUIVALENCE (KAVEL,NAVEL)
15.     EQUIVALENCE (KCENLAT,NCENLAT)
16.     EQUIVALENCE (KCENEL,NCENEL)
17.     EQUIVALENCE (KCENFA,NCENFA)
18.     EQUIVALENCE (KCENLONG,NCENLONG)
19.     DATA INIT/0000/
20.     DATA END/'EITP'/
21.     IF(INIT.NE.0) GO TO 200
22.     OUTPUT 'SUBROUTINE GETGC VERSION 10 JAN 76'
23.     INIT=1
24.     C      INITIALIZATION
25.     100 CONTINUE
26.     NCARD=105
27.     NDLT=100
28.     KURLAT=999
29.     KURLONG=999
30.     NRET=0
31.     IE9D=0
32.     DO 107 I=1,10
33.     DO 106J=1,10
34.     NPRINT=108
35.     KAVFA(I,J)=0
36.     KAVEL(I,J)=0
37.     NPTS(I,J)=0
38.     KCENEL(I,J)=0
39.     KCENFA(I,J)=0
40.     H0WFAR(I,J)=999
41.     KCENLAT(I,J)=0
42.     KCENLONG(I,J)=0
43.     106 CONTINUE
44.     107 CONTINUE
45.     NRET=0
46.     ILAST=0
47.     NBR=0
48.     NI=0
49.     RADEG=57.2958
50.     DEGRA=1.7453E-2
51.     IF(ISW(25).EQ.0) GO TO 120
52.     READ(NCARD,1001) NTAPESN
53.     1001 FORMAT(20A4)
54.     IF(NTAPESN.EQ.'EITP') GO TO 999
55.     CALL MBUNT(ITAPE,NTAPESN)
56.     WRITE(NPRINT,1002) NTAPESN
57.     1002 FORMAT(1X,/,5X,'TAPE S/N',A4, ' MBUNTED')
58.     120 CONTINUE
59.     KDTOP=RTOP*RADEG+200.

```

```

60.      KDBBT=RBBT*RADEG+200.
61.      KDLFT=RLFT*RADEG+200.
62.      KDRGT=RRGT*RADEG+200.
63.      110 CONTINUE
64.      CALL BUFFER IN(ITAPE,0,IBUFIN(1,1),1650,IKEY,NI)
65.      GO TO (111,115,113,114) IKEY
66.      111 OUTPUT 'WAITING'; GO TO 110
67.      113 OUTPUT 'END OF FILE ON INPUT'; GO TO 900
68.      114 OUTPUT 'ERROR ON INPUT'; STOP
69.      115 CONTINUE
70.      NI=(NI*4)/22
71.      DECODE(NI*22,1005,IBUFIN(1,1),ND)
72.      * (KODE(I),KDLAT(I),KDLANG(I),KELDER(I),KFA(I),I=1,NI)
73.      IDECODE=0
74.      NIN=0
75.      IF(NX.EQ.4) GO TO 200
76.      IF(NX.EQ.5) GO TO 200
77.      IF(NX.EQ.6) GO TO 200
78.      IF(NX.EQ.7) GO TO 200
79.      IF(NX.GE.8.AND.NRET.EQ.0) GO TO 200
80.      IF(NX.GE.8.AND.NRET.GT.0) GO TO 665
81.      200 CONTINUE
82.      1010 FFORMAT (10(1X,I6))
83.      1011 FFORMAT(1X,I5,4A4)
84.      IF(ILAST.EQ.1) GO TO 910
85.      IF(NRET.GT.0) GO TO 665
86.      IF(ISW(46).EQ.1) GO TO 205
87.      IF(NBR.GT.0) GO TO 205
88.      IF(NBR.EQ.0) READ(NDLT,1003,END=900) NBR,NDLTTL,NDLTLGR,NDLTLGL
89.      1003 FFORMAT(5X,I6,3I3)
90.      C      CHECK BOUNDS
91.      IF(NDLTTL.GT.KDTAP) GO TO 500
92.      IF((NDLTLGL.LE.KDRGT).AND.(NDLTLGR.GE.KDLFT)) GO TO 204
93.      GO TO 500
94.      204 CONTINUE
95.      IF(NDLTTL.LT.KDBBT) GO TO 500
96.      C      WITHIN BOUNDS
97.      205 CONTINUE
98.      IF(NIN.LT.NI) GO TO 220
99.      210 CONTINUE
100.     CALL BUFFER IN(ITAPE,0,IBUFIN(1,1),1650,IKEY,NI)
101.     GO TO (211,215,213,214) IKEY
102.     211 OUTPUT 'WAITING'; GO TO 210
103.     213 OUTPUT 'END OF FILE ON INPUT'; GO TO 900
104.     214 OUTPUT 'ERROR ON INPUT'; STOP
105.     215 CONTINUE
106.     NI=(NI*4)/22
107.     DECODE(NI*22,1005,IBUFIN(1,1),ND)
108.     * (KODE(I),KDLAT(I),KDLANG(I),KELDER(I),KFA(I),I=1,NI)
109.     IDECODE=0
110.     1005 FFORMAT(300(11,215,I6,I5))
111.     NIN=0
112.     220 CONTINUE
113.     IF(IDECODE.EQ.1)
114.     *DECODE(NI*22,1005,IBUFIN(1,1),ND)
115.     * (KODE(I),KDLAT(I),KDLANG(I),KELDER(I),KFA(I),I=1,NI) ;
116.     * IDECODE=0
117.     NIN=NIN+1
118.     NBR=NBR+1
119.     KLAT=KDLAT(NIN)/100

```

```

120.      KLONG=KDLONG(NIN)/100
121.      IF(ISW(46).EQ.1) GO TO 230
122.      IF((KLAT.GT.KDTBP).OR.(KLAT.LT.KDBBT)) GO TO 200
123.      IF((KLONG.LT.KDLFT).OR.(KLONG.GT.KDRGT)) GO TO 200
124.      230 CONTINUE
125.      GO TO (240,250,260,270,600,600,600,600,600,600)(NX-
126.      C      DEPTH
127.      240 CONTINUE
128.      IF(KELDEP(NIN).GE.0) GO TO 200
129.      DATA=-FLBAT(KELDEP(NIN))
130.      GO TO 480
131.      C      ELEVATION
132.      250 CONTINUE
133.      IF(KELDEP(NIN).LT.0) GO TO 200
134.      DATA=FLBAT(KELDEP(NIN))
135.      GO TO 480
136.      C      FREE AIR
137.      260 CONTINUE
138.      DATA=FLBAT(KFA(NIN)/10)
139.      GO TO 480
140.      C      BUGGER
141.      270 CONTINUE
142.      OUTPUT 'BUGURE CALC NOT IMPLEMENTED';STOP
143.      280 CONTINUE
144.      IF(NY.EQ.0) GO TO 480
145.      GO TO (340,350,360,370)(NY-3)
146.      340 CONTINUE
147.      IF(KELDEP(NIN).GE.0) GO TO 200
148.      DATA=-FLBAT(KELDEP(NIN))
149.      GO TO 380
150.      350 CONTINUE
151.      IF(KELDEP(NIN).LT.0) GO TO 200
152.      DATA=FLBAT(KELDEP(NIN))
153.      GO TO 380
154.      360 CONTINUE
155.      DATA=FLBAT(KFA(NIN)*10)
156.      GO TO 380
157.      370 CONTINUE
158.      380 CONTINUE
159.      IF(NZ.EQ.0) GO TO 480
160.      GO TO (440,450,460,470)(NZ-3)
161.      440 CONTINUE
162.      IF(KELDEP(NIN).GE.0) GO TO 200
163.      DATA=-FLBAT(KELDEP(NIN))
164.      GO TO 480
165.      450 CONTINUE
166.      IF(KELDEP(NIN).LT.0) GO TO 200
167.      DATA=FLBAT(KELDEP(NIN))
168.      GO TO 480
169.      460 CONTINUE
170.      DATA=FLBAT(KFA(NIN)*10)
171.      GO TO 480
172.      470 CONTINUE
173.      480 CONTINUE
174.      RLAT=((FLBAT(KDLAT(NIN))/100.)-200.)*DEGRA
175.      RLONG=((FLBAT(KDLONG(NIN))/100.)-200.)*DEGRA
176.      RETURN
177.      500 CONTINUE
178.      C      CHECK IF PAST BOUNDS
179.      IF((NDLTGR.LT.KDLFT).AND.(NDLTLT.LT.KDBBT))

```

```

180.      * OUTPUT 'PAST BOUNDS' ; GO TO 900
181. C      OUTSIDE BOUNDS
182.      502 CONTINUE
183.      IF(NBR.LT.(300-NIN))
184.      * IDECBDE=1 ;
185.      *      NIN=NIN+NBR; NBR=0; GO TO 200
186.      NBR=NBR-(300-NIN)
187.      CALL BUFFER IN(ITAPE,0,IBUFIN(1,1),1650,IKEY,NI)
188.      1012 FORMAT(1X,2I5,3A4)
189.      NI=(NI*4)/22
190.      505 CONTINUE
191.      GO TO(511,515,513,514) IKEY
192.      511 OUTPUT 'WAITING' ; GO TO 505
193.      513 OUTPUT 'EOF FOUND WHILE SKIPPING'; GO TO 900
194.      514 OUTPUT 'ERROR WHILE SKIPPING'
195.      515 CONTINUE
196.      NIN=0
197.      GO TO 502
198. C      ABSTRACT DATA BEFORE PLOTTING
199.      600 CONTINUE
200.      IF(KURLAT.EQ.999) GO TO 668
201.      IF((KLAT.EQ.KURLAT).AND.(KLONG.EQ.KURLONG)) GO TO 700
202. C      NEW DEGREE SQUARE
203. C
204. C      PREPARING PRIOR DEGREE SQUARE FOR OUTPUT
205.      DO 660 I=1,10
206.      DO 650 J=1,10
207.      KAVFA(I,J)=KAVFA(I,J)/NPTS(I,J)
208.      KAVEL(I,J)=KAVEL(I,J)/NPTS(I,J)
209.      650 CONTINUE
210.      660 CONTINUE
211.      665 CONTINUE
212.      NRET=NRET+1
213.      IF(NRET.GT.100) GO TO 668
214.      IF(NX.EQ.3) DATA=FLBAT(NCENFA(NRET))*0.1; GO TO 667
215.      IF(NX.EQ.9) DATA=FLBAT(NCENEL(NRET)); GO TO 667
216.      IF(NX.EQ.12) DATA=FLBAT(NAVFA(NRET))*0.1; GO TO 667
217.      IF(NX.EQ.13) DATA=FLBAT(NAVEL(NRET)) ; GO TO 667
218.      667 CONTINUE
219.      IF(DATA.EQ.0) GO TO 665
220.      RLAT=((FLBAT(NCENLAT(NRET))/100.)-200.)*DEGRA
221.      RLONG=((FLBAT(NCENLONG(NRET))/100.)-200.)*DEGRA
222.      RETURN
223.      668 CONTINUE
224.      KURLAT=KLAT
225.      KURLONG=KLONG
226.      NRET=0
227. C      CLEARING ARRAYS BEFORE NEW DEGREE SQUARE
228.      DO 680 I=1,10
229.      DO 670 J=1,10
230.      KAVFA(I,J)=0
231.      KAVEL(I,J)=0
232.      NPTS(I,J)=0
233.      KCENEL(I,J)=0
234.      KCENFA(I,J)=0
235.      HCNFA(I,J)=999.0
236.      KCENLAT(I,J)=0
237.      KCENLONG(I,J)=0
238.      670 CONTINUE
239.      680 CONTINUE

```

```

240.      C          SAME DEGREE SQUARE
241.      700 CONTINUE
242.          DLAT=FLOAT(KDLAT(NIN))/100.
243.          DLONG=FLOAT(KDLONG(NIN))/100.0
244.          DECLAT=DLAT-FLOAT(IFIX(DLAT))+0.001
245.          DECLONG=DLONG-FLOAT(IFIX(DLONG))+0.001
246.          KDECLAT=IFIX(DECLAT*100.)
247.          KDECLONG=IFIX(DECLONG*100.)
248.          I=IFIX(DECLAT*10.)+1
249.          J=IFIX(DECLONG*10.)+1
250.          KAVEL(I,J)=KAVEL(I,J)+KELDEP(NIN)
251.          KAVFA(I,J)=KAVFA(I,J)+KFA(NIN)
252.          NPTS(I,J)=NPTS(I,J)+1
253.          DIST=SQRT((KDECLAT-((I-1)*10+5))**2+(KDECLONG-((J-1)*10+5))**2)
254.          IF(DIST.LT.HOWFAR(I,J)) HOWFAR(I,J)=DIST,
255.          *      KCENLAT(I,J)=KDLAT(NIN),
256.          *      KCENLONG(I,J)=KDLONG(NIN),
257.          *      KCENFA(I,J)=KFA(NIN),
258.          *      KCENEL(I,J)=KELDEP(NIN)
259.          GO TO 200
260.      C          PREPARING LAST DEGREE SQUARE
261.      C          END OF FILE
262.      900 CONTINUE
263.          IF(NX.LT.8) GO TO 999
264.          ILAST=1
265.          DO 906 I=1,10
266.          DO 905 J=1,10
267.              KAVFA(I,J)=KAVFA(I,J)/NPTS(I,J)
268.              KAVEL(I,J)=KAVEL(I,J)/NPTS(I,J)
269.      905 CONTINUE
270.      906 CONTINUE
271.      C          BUTPUT LAST DEGREE SQUARE
272.      910 CONTINUE
273.          NRET=NRET+1
274.          IF(NRET.GT.100) GO TO 999
275.          IF(NX.EQ.8) DATA=FLOAT(KCENFA(NRET))*0.1, GO TO 967
276.          IF(NX.EQ.9) DATA=FLOAT(KCENEL(NRET)), GO TO 967
277.          IF(NX.EQ.12) DATA=FLOAT(KAVFA(NRET))*0.1, GO TO 967
278.          IF(NX.EQ.13) DATA=FLOAT(KAVEL(NRET)) , GO TO 967
279.      967 CONTINUE
280.          IF(DATA.EQ.0) GO TO 910
281.          RLAT=((FLOAT(KCENLAT(NRET))/100.)*200.)*DEGRA
282.          RLONG=((FLOAT(KCENLONG(NRET))/100.)*200.)*DEGRA
283.          RETURN
284.      C          END OF JOB
285.      999 CONTINUE
286.          IEED=1
287.          RETURN
288.          END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
BUFFERIN	R	SPRG	01030	V DUMMY	DATA	R	SCALR	0102C	V DUMMY	DATA	R	SCALR	0102E	V DUMMY	DECLBNG	R	SCALR	01026	V DUMMY
DEGRA	R	SCALR	01013	V 1	DECLAT	R	SCALR	01025	V 1	DECLBNG	R	SCALR	01028	V 1	DLAT	R	SCALR	01023	V 1
DLNG	R	SCALR	01024	V 1	ELTP	R	SCALR	01015	V 1	END	R	SCALR	01006	V 1	GETGC	R	SPRG	00000	P
FLGAT	R	SPRG	INTNIN		GETGC	R	SCALR	00000	V 1	IBUFIN	I	ARRAY	00001	V 1800	IFIX	I	SPRG	INTRIN	
HWFAF	R	ARRAY	00E11	V 100	IFIX	I	SCALR	0100C	V 1	INIT	I	SCALR	01005	V 1	J	I	SCALR	0100D	V 1
IDECRDE	I	SCALR	0101C	V 1	ILAST	I	SCALR	0100F	V 1	KCENEL	I	ARRAY	00F3D	V 100	KCENLNG	I	ARRAY	00ED9	V 100
IKEY	I	SCALR	0101A	V 1	ITAPE	I	SCALR	0102A	V DUMMY	KDECLBNG	I	SCALR	00E75	V 100	KDECLNG	I	SCALR	00E28	V 1
ISW	I	SPRG	EXTERN		KAVFA	I	ARRAY	00CE5	V 100	KELDEP	I	ARRAY	00961	V 300	KLNG	I	SCALR	01022	V 1
KAVEL	I	ARRAY	00U49	V 100	KDECLAT	I	SCALR	01027	V 1	KLNG	I	SCALR	01021	V 1	NAVFA	I	SCALR	0100A	V 1
KCENFA	I	ARRAY	00FA1	V 100	KDLFT	I	SCALR	01018	V 1	KURLNG	I	SCALR	00CE5	V 100	NCENEL	I	ARRAY	00F3D	V 100
KDBBT	I	SCALR	01017	V 1	KDTBP	I	SCALR	01016	V 1	NCENLNG	I	ARRAY	00ED9	V 100	NDLTGL	I	SCALR	01020	V 1
KDLAT	I	ARRAY	00835	V 300	KLAT	I	SCALR	01021	V 1	NPTS	I	SCALR	01011	V 1	RADEG	R	SCALR	01012	V DUMMY
KDRGT	I	SCALR	01019	V 1	KURLAT	I	SCALR	01009	V 1	RLFT	R	SCALR	01033	V DUMMY	RTBP	R	SCALR	01031	V DUMMY
KFA	I	ARRAY	00889	V 300	NAVEL	I	SCALR	00D49	V 100										
KRDE	I	ARRAY	00709	V 300	NCARD	I	SCALR	01007	V 1										
KRUNT	I	SPRG	EXTERN		NCENLAT	I	ARRAY	00E75	V 100										
KR	I	SCALR	01010	V 1	NDLT	I	SCALR	01008	V 1										
NENFA	I	SCALR	01018	V 1	NDLTGL	I	SCALR	0101E	V 1										
ND	I	SCALR	0101F	V 1	NPRINT	I	SCALR	0100E	V 1										
NDLTGLR	I	SCALR	0101D	V 1	NZ	I	SCALR	0102F	V DUMMY										
NIN	I	SCALR	0100B	V 1	NTAPEN	I	SCALR	01034	V DUMMY										
NRET	I	SCALR	01020	V DUMMY	KLAT	R	SCALR	01035	V DUMMY										
NR	I	SCALR	01032	V DUMMY	KRGT	R	SCALR	01034	V DUMMY										
NR9T	R	SCALR	01036	V DUMMY															
PLNG	R	SCALR	INTNIN																
SRGT	R	SPRG																	

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
00000	GETGC		00001	IBUFIN	00000	GETGC		00001	IBUFIN	00000	GETGC		00001	IBUFIN	00000	GETGC		00001	IBUFIN
00009	KFA		00005	KAVFA	00009	KFA		00005	KAVFA	00009	KFA		00005	KAVFA	00009	KFA		00005	KAVFA
00011	HWFAF		00007	NCENLAT	00011	HWFAF		00007	NCENLAT	00011	HWFAF		00007	NCENLAT	00011	HWFAF		00007	NCENLAT
00030	KCENEL		00013	NCENFA	00030	KCENEL		00013	NCENFA	00030	KCENEL		00013	NCENFA	00030	KCENEL		00013	NCENFA
00033	NDLT		00014	KURLAT	00033	NDLT		00014	KURLAT	00033	NDLT		00014	KURLAT	00033	NDLT		00014	KURLAT
00036	NPRINT		00015	ILAST	00036	NPRINT		00015	ILAST	00036	NPRINT		00015	ILAST	00036	NPRINT		00015	ILAST
00039	IKEY		00016	ELTP	00039	IKEY		00016	ELTP	00039	IKEY		00016	ELTP	00039	IKEY		00016	ELTP
00042	NDLTGL		00017	KLAT	00042	NDLTGL		00017	KLAT	00042	NDLTGL		00017	KLAT	00042	NDLTGL		00017	KLAT
00045	NR		00018	NAVEL	00045	NR		00018	NAVEL	00045	NR		00018	NAVEL	00045	NR		00018	NAVEL
00048	NENFA		00019	NCENLAT	00048	NENFA		00019	NCENLAT	00048	NENFA		00019	NCENLAT	00048	NENFA		00019	NCENLAT
00051	ND		00020	NTAPEN	00051	ND		00020	NTAPEN	00051	ND		00020	NTAPEN	00051	ND		00020	NTAPEN
00054	NDLTGLR		00021	KLAT	00054	NDLTGLR		00021	KLAT	00054	NDLTGLR		00021	KLAT	00054	NDLTGLR		00021	KLAT
00057	NIN		00022	NAVEL	00057	NIN		00022	NAVEL	00057	NIN		00022	NAVEL	00057	NIN		00022	NAVEL
00060	NRET		00023	NCARD	00060	NRET		00023	NCARD	00060	NRET		00023	NCARD	00060	NRET		00023	NCARD
00063	NR		00024	NCENLAT	00063	NR		00024	NCENLAT	00063	NR		00024	NCENLAT	00063	NR		00024	NCENLAT
00066	NR9T		00025	NDLT	00066	NR9T		00025	NDLT	00066	NR9T		00025	NDLT	00066	NR9T		00025	NDLT
00069	PLNG		00026	KLAT	00069	PLNG		00026	KLAT	00069	PLNG		00026	KLAT	00069	PLNG		00026	KLAT
00072	SRGT		00027	NAVEL	00072	SRGT		00027	NAVEL	00072	SRGT		00027	NAVEL	00072	SRGT		00027	NAVEL
00075			00028	NCENLAT	00075			00028	NCENLAT	00075			00028	NCENLAT	00075			00028	NCENLAT
00078			00029	NDLT	00078			00029	NDLT	00078			00029	NDLT	00078			00029	NDLT
00081			00030	NTAPEN	00081			00030	NTAPEN	00081			00030	NTAPEN	00081			00030	NTAPEN
00084			00031	KLAT	00084			00031	KLAT	00084			00031	KLAT	00084			00031	KLAT
00087			00032	NAVEL	00087			00032	NAVEL	00087			00032	NAVEL	00087			00032	NAVEL
00090			00033	NCENLAT	00090			00033	NCENLAT	00090			00033	NCENLAT	00090			00033	NCENLAT
00093			00034	NDLT	00093			00034	NDLT	00093			00034	NDLT	00093			00034	NDLT
00096			00035	KLAT	00096			00035	KLAT	00096			00035	KLAT	00096			00035	KLAT
00099			00036	NAVEL	00099			00036	NAVEL	00099			00036	NAVEL	00099			00036	NAVEL
00102			00037	NCENLAT	00102			00037	NCENLAT	00102			00037	NCENLAT	00102			00037	NCENLAT
00105			00038	NDLT	00105			00038	NDLT	00105			00038	NDLT	00105			00038	NDLT
00108			00039	NTAPEN	00108			00039	NTAPEN	00108			00039	NTAPEN	00108			00039	NTAPEN
00111			00040	KLAT	00111			00040	KLAT	00111			00040	KLAT	00111			00040	KLAT
00114			00041	NAVEL	00114			00041	NAVEL	00114			00041	NAVEL	00114			00041	NAVEL
00117			00042	NCENLAT	00117			00042	NCENLAT	00117			00042	NCENLAT	00117			00042	NCENLAT
00120			00043	NDLT	00120			00043	NDLT	00120			00043	NDLT	00120			00043	NDLT
00123			00044	KLAT	00123			00044	KLAT	00123			00044	KLAT	00123			00044	KLAT
00126			00045	NAVEL	00126			00045	NAVEL	00126			00045	NAVEL	00126			00045	NAVEL
00129			00046	NCENLAT	00129			00046	NCENLAT	00129			00046	NCENLAT	00129			00046	NCENLAT
00132			00047	NDLT	00132			00047	NDLT	00132			00047	NDLT	00132			00047	NDLT
00135			00048	NTAPEN	00135			00048	NTAPEN	00135			00048	NTAPEN	00135			00048	NTAPEN
00138			00049	KLAT	00138			00049	KLAT	00138			00049	KLAT	00138			00049	KLAT
00141			00050	NAVEL	00141			00050	NAVEL	00141			00050	NAVEL	00141			00050	NAVEL
00144			00051	NCENLAT	00144			00051	NCENLAT	00144			00051	NCENLAT	00144			00051	NCENLAT
00147			00052	NDLT	00147			00052	NDLT	00147			00052	NDLT	00147			00052	NDLT
00150			00053	KLAT	00150			00053	KLAT	00150			00053	KLAT	00150			00053	KLAT
00153			00054	NAVEL	00153			00054	NAVEL	00153			00054	NAVEL	00153			00054	NAVEL
00156			00055	NCENLAT	00156			00055	NCENLAT	00156			00055	NCENLAT	00156			00055	

01026 DECLONG    01027 KDECLAT    01028 KDECLONG    01029 DIST

BLANK COMMON (0 WORDS)

ENTRY POINTS:

00000 GETGC

INTRINSIC SUBPROGRAMS USED:

FL0AT    IFIX    SQRT

EXTERNAL SUBPROGRAMS REQUIRED:

BUFFERIN	ISW	MBUNT	F:101	F:102	F:103	F:104	F:105
F:106	F:108	9BCDRDEE	9BCDREAD	9BCDWRT	9DEC0DE	9ENDI0L	9IBDATA
9IT0R	9PRINT	9RT0I	9SETUPN	9SQRT	9ST0P		

HIGHEST ERROR SEVERITY: 0 (NR ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	997	003E5
CONSTANTS:	16	00010
LOCAL VARIABLES:	4138	0102A
TEMPS:	16	00010
TOTAL PROGRAM:	5167	0142F

```

1. SUBROUTINE GETGS(ITAPE,NX,DATA,X,FM,T,RT,RF,RR,BT,RLEFT,RRIGT,
2. 1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHM,IEBC,IBLK)
3. C   VERSION 30 JAN 75, CORRECT READ FOR ISW(29)
4. C   VERSION 26 JUNE 74, FINAL TOUCHES CONVERSION TO 67 G FORMULA
5. C   VERSION 12 JUNE 1974
6. C   VERSION OF 10 AUGUST 1973, START CONVERSION TO 1967 G FORMULA
7. C   VERSION 30 MAY 73
8. C
9. C
10. C   SSW(12) UP TO LIST DATE IDENTIFICATION
11. C   SSW(27) UP TO SUPPRESS REWIND OF TAPES AT START OF JOB
12. C   = 1 SUPPRESS REWIND OF ITAPE
13. C   SSW(29) = 1 - TO READ AND TEST FOR SELECTED SOURCE CODE
14. C   NUMBERS TO BE PROCESSED
15. C   = 2 - TO READ AND TEST FOR SELECTED SOURCE CODE
16. C   NUMBERS TO BE SKIPPED
17. C
18. C
19. C   SSW(40) UP TO PROCESS WITH BOUNDS USING CLT
20. C   = 0 - PROCESS WITHOUT BOUNDS
21. C   = 1 - PROCESS WITH BOUNDS USING THE DATA LOCATION TABLE
22. C
23. C   USES ENDIO(DUMMY), EVIL, STAT, ISW
24. C   ASSUME ISW AND STAT INITIALIZED IN MAIN PROGRAM
25. C
26. C
27. C   DIMENSION IGSL(40),IFMT(40)
28. C
29. C   DIMENSION BLF(50,32),BLI(128),BLK(128)
30. C   DIMENSION IS(20,5),IL(20,5)
31. C   DIMENSION ILBN(20),IPIE(20)
32. C   DIMENSION NVAR(20),IVFM(20,5)
33. C
34. C   DIMENSION RVAR(5)
35. C   DIMENSION IDTIN(20),IBK(20),ITK(20),IDENS(20),IDESC(17,20)
36. C   DIMENSION IDTOT(20),JBK(20),JTK(20),JDENS(20),JDESC(17,20)
37. C   DIMENSION PLT(20),IA(35),IAFMT(9),IASH(35),ISRC(16)
38. C   DATA ITERI/'EITP'/
39. C   DATA IBL/' '/
40. C   DATA IFLAG/0/,IFIN/0/,IFLGI/0/
41. C   DATA IGSL/'(1,14,312,14,2F9.4,13,13)
42. C
43. C
44. C   DATA NVAR(3),ILBN(3),IPIE(3),
45. C   * (IVFM(3,0),,=1,2),IS(3,1),IL(3,1)
46. C   * /1,2,1,'F7.2, '34,7/
47. C   DATA NVAR(4),ILBN(4),IPIE(4),
48. C   * (IVFM(4,0),,=1,2),IS(4,1),IL(4,1)
49. C   * /1,2,1,'F5.0, '50,5/
50. C   DATA NVAR(5),ILBN(5),IPIE(5),
51. C   * (IVFM(5,0),,=1,3),IS(5,1),IL(5,1),
52. C   * IS(5,2),IL(5,2)
53. C   * /2,3,2,'F7.2,F5.0, '34,7,50,5/
54. C   DATA NVAR(6),ILBN(6),IPIE(6),
55. C   * (IVFM(6,0),,=1,2),IS(6,1),IL(6,1)
56. C   * /1,2,1,'F6.1, '61,6/
57. C   DATA NVAR(7),ILBN(7),IPIE(7),
58. C   * (IVFM(7,0),,=1,2),IS(7,1),IL(7,1)
59. C   * /1,2,1,'F6.1, '61,6/

```

GIN80140  
GIN80150

GIN80240

GIN80270  
GIN80280  
GIN80290  
GIN80300  
GIN80310

GIN80320  
GIN80330

```

60.      DATA NVAR(8), IL6N(8), IPIE(8),
61.      * (IVFM(8,J), J=1,2), IS(8,1), IL(8,1)
62.      * /1,2,1,'F4.1', '67,4/
63.      DATA NVAR(9), IL6N(9), IPIE(9),
64.      * (IVFM(9,J), J=1,3), IS(9,1), IL(9,1),
65.      * IS(9,2), IL(9,2)
66.      * /2,3,2,'F6.1,F4.1', '61,6,67,4/
67.      DATA NVAR(10), IL6N(10), IPIE(10),
68.      * (IVFM(10,J), J=1,2), IS(10,1), IL(10,1)
69.      * /1,2,1,'F6.1', '75,6/
70.      DATA NVAR(11), IL6N(11), IPIE(11),
71.      * (IVFM(11,J), J=1,3), IS(11,1), IL(11,1)
72.      * /2,3,1,'F3.0,F6.2', '34,9/
73.      C
74.      C
75.      DATA NVAR(13), IL6N(13), IPIE(13),
76.      * (IVFM(13,J), J=1,3), IS(13,1), IL(13,1), IS(13,2), IL(13,2)
77.      * /5,3,2,'4F2.C,F8.1', '82,8,90,8/
78.      DATA NVAR(14), IL6N(14), IPIE(14),
79.      * (IVFM(14,J), J=1,3), IS(14,1), IL(14,1), IS(14,2), IL(14,2)
80.      * /5,3,2,'4F2.C,F6.1', '82,8,98,6/
81.      DATA NVAR(15), IL6N(15), IPIE(15),
82.      * (IVFM(15,J), J=1,3), IS(15,1), IL(15,1), IS(15,2), IL(15,2)
83.      * /5,3,2,'4F2.C,F6.1', '82,8,104,6/
84.      DATA NVAR(16), IL6N(16), IPIE(16),
85.      * (IVFM(16,J), J=1,3), IS(16,1), IL(16,1), IS(16,2), IL(16,2)
86.      * /5,3,2,'4F2.C,F4.0', '82,8,110,4/
87.      DATA NVAR(17), IL6N(17), IPIE(17),
88.      * (IVFM(17,J), J=1,3), IS(17,1), IL(17,1), IS(17,2), IL(17,2)
89.      * /5,3,2,'4F2.C,F4.1', '82,8,115,4/
90.      C
91.      DIMENSION IST(20), ILT(20)
92.      DATA IST(1), ILT(1), IST(2), ILT(2) /1,33,121,6/
93.      C
94.      C
95.      IF (IFLAG.NE.C) GOT050
96.      IFLAG=1
97.      C
98.      C
99.      C ----- GETG INITIALISATION LOGIC
100.     C
101.     ILI= 99 ; IDL=100
102.     ICND=C ; ICHA=C ; ICNT=C
103.     IIN=105
104.     IIBLT = 108
105.     IEED=0
106.     C
107.     C
108.     KMAX=39 ; KN=2
109.     C
110.     DEGRA = 1.745329E-2
111.     RADEG=57.29578
112.     C
113.     CLAB8=RBOT*RADEG ; DLAT8=RTOP*RADEG
114.     CL0LE=RLEFT*RADEG ; CL0RI=RRIGT*RADEG
115.     C
116.     NZER8=0
117.     KGDA8=NZER8
118.     KGM08=NZER8
119.     KGYR8=NZER8

```

GIN80390

GIN80410

GIN80430

GIN80470

GIN80480

GIN80490

GIN80500

```

120.      KGHM8=NZER8
121. C      NRECT = N8. 8F RECORDS NOW WRITTEN ON PRESENT OUTPUT TAPE
122. C      NEF = N8. 8F FILE NOW BEING PROCESSED
123.      NRECT = NZER8
124.      IREC1 = 1
125. C
126.      BUTPLT 'GETGS VERSION 3C JAN 75 FOR 67 C FORMULA'
127. C
128. C ----- CHECK SSW(29) TO SEE IF SOURCE CODE NUMBERS
129. C ----- ARE TO BE READ FOR DATA SELECTION
130. C
131.      IF(ISW(29).EG.C)G0T01405
132.      READ(IIN,900)ISRC
133.      900 FORMAT(16I5)
134.      IF(ISW(29).EG.1)WRITE(IIBLT,912)ISRC ; G0T01405
135.      WRITE(IIBLT,913)ISRC
136.      913 FORMAT(1H0,10X,'SKIPPED SOURCE CODES = ',16I5)
137.      912 FORMAT(1H0,10X,'SELECTED SOURCE CODES = ',16I5)
138. C
139. C -----
140. C
141.      1405 IF(ISW(40).NE.C)CALL ENDT(J,CLAT8,CLAB8,CL0LE,CL0RI,IDL,ILI,0)
142.      * ; CALL SETSKP(INDICA) ; IDLT=0
143.      * ; NEF=1 ; IFILE=J-1 ; BUTPUT NEF,IFILE
144.      * ; G0 T0 1410
145.      IF(ISW(30))404,404,1410
146.      404 J=1
147.      405 READ (IIN,406) IDTIN(J),IBK(J),ITK(J),IDENS(J),
148.      1 (IDESC(K,J),K=1,17)
149.      406 FORMAT(A4,1X,A1,1X,I1,1X,I3,17A4)
150.      IF(IDTIN(J).NE.ITERI) J=J+1; G0 T0 405
151.      NEF=1
152.      IFILE=J - 1
153.      BUTPUT NEF,IFILE
154. C
155. C
156.      1410 IF(ISW(30))10,10,2414
157.      10 IF(ISW(40).EG.C)G0T0811
158.      READ(IDL,406)IDTIN(1),IBK(1),ITK(1),IDENS(1),
159.      * (IDESC(K,1),K=1,17) ; IMDL=C
160.      811 CALL MBLNT(ITAPE,IDTIN(1))
161.      BUTPLT 'INPUT TAPE MOUNTED '
162.      WRITE (IIBLT,1413) IDTIN(1),IBK(1),ITK(1),IDENS(1),
163.      1 (IDESC(K,1),K=1,17)
164.      1413 FORMAT (1X,A4,1X,A1,1X,I1,1X,I3,17A4)
165.      810 IF(IDTIN(1).EG.ITERI)IE0D=1 ; RETURN
166.      BUTPLT '-----'
167.      IF(ISW(27).EG.1)G0T02414
168.      REWIND ITAPE
169.      IF((FMT.NE.3)PRINT920 ; IE0D=1 ; RETURN
170.      920 FORMAT(1H0,5X,'UFMT NOT EQUAL TO 3 , GETG CAN ONLY PROCESS GSUM
171.      * DATA,/)
172.      D0924I=1,40
173.      IFMT(I)=IGSU(I)
174.      924 CONTINUE
175.      IF(NX.LT.3)G0T02414
176.      IF((NX.EG.12).OR.(NX.GT.17)) PRINT921,NX,IE0D=1; RETURN
177.      921 FORMAT(1H0,5X,'NX',3X,I2,3X,' THIS OPTION NOT YET INCLUDED')
178. C
179.      IFMS1=IFMT(6) ; IFMS2=IFMT(7)

```

GIN00510  
GIN00520  
GIN00530  
GIN00540  
GIN00550  
GIN00560

GIN00580  
GIN00590  
GIN00600  
GIN00610  
GIN00620  
GIN00630  
GIN00640  
GIN00650

GIN00780  
GIN00790  
GIN00800

GIN00820

```

180.      C0922I=1,IL0N(NX)
181.      IKSL=I+5 ; IFMT(IKSL)=IVFM(NX,I)
182.      922 CONTINUE
183.      IKSL=IKSL+1 ; IFMT(IKSL)=IFMS1
184.      IKSL=IKSL+1 ; IFMT(IKSL)=IFMS2
185.      C
186.      IST1=IST(2) ; ILT1=ILT(2)
187.      C0923I=1,IPIE(NX)
188.      KN=KN+1
189.      IKSL=I+1 ; IST(IKSL)=IS(NX,I) ; ILT(IKSL)=IL(NX,I)
190.      KMAX=KMAX+IL(NX,I)
191.      923 CONTINUE
192.      IKSL=IKSL+1 ; IST(IKSL)=IST1 ; ILT(IKSL)=ILT1
193.      C
194.      2414 CONTINUE
195.      PRINT2415,IFMT
196.      2415 FORMAT(1H0,5X,'RUN TIME FORMAT = ',//,1X,20A4,/,1X,20A4,/,
197.      * 6X,'-----',//)
198.      C
199.      C ----- GETG INPUT LOGIC
200.      C
201.      50 CALL ENDIG
202.      IF(ISW(40).NE.C)G0T0501
203.      C
204.      52 CALL BLFIG(ITAPE,IEND,1,C,
205.      *IBLK,32,KMAX,KN,IST,ILT,BLF,BLI,BLK,ICNT)
206.      IF(IEND.EG.1)I=2 ; G0 T0 710
207.      C
208.      IF(NX.GT.2)G0T0950
209.      DECODE(KMAX,IFMT,BLI,IREF1,IS0RC,KGDA,KGM0,KGYR,KGHM,DLAT,DL0NG,
210.      * LTKEY,LGKEY
211.      IF(NX.EG.1)VAR=KGHM ; G0T0970
212.      IF(NX.EG.2)VAR=IS0RC
213.      G0T0970
214.      950 DECODE(KMAX,IFMT,BLI,IREF1,IS0RC,KGDA,KGM0,KGYR,KGHM,DLAT,DL0NG,
215.      * (RVAR(J),J=1,NVAR(NX)),
216.      * LTKEY,LGKEY
217.      IF(IFLGI.EG.1) G0 T0 373
218.      IFLGI = 1
219.      WRITE(IIBLT,374) IREF1
220.      374 FORMAT(' INPUT IREF = ',I2)
221.      IF (IREF1.EG.1) OUTPLT ' CONVERTING TO 67 FORMULA IN THIS RUN.
222.      IF (IREF1.EG.2) OUTPLT ' INPUT ALREADY IN 67 FORMULA'
223.      373 CONTINUE
224.      IF(NX.NE.5)G0T0953
225.      IF(RVAR(2).EG.C)VAR=RVAR(1) ; G0T0970
226.      VAR=RVAR(2) ; G0T0970
227.      953 IF(NX.NE.9)G0T0954
228.      VAR=RVAR(1)*RVAR(2) ; G0T0970
229.      954 IF(NX.NE.11)G0T0955
230.      C NX = 11 TO PLOT OBSERVED GRAVITY
231.      C THIS ROUTINE CANNOT PLOT OBSERVED GRAVITY
232.      A=RVAR(1)-977. ; A=A*1000.0
233.      621 VAR=RVAR(2)*A ; G0T0970
234.      955 IF(NX.LT.13)G0T0956
235.      IF(RVAR(1).NE.10.)PRINT957 ; IE00=1 ; RETURN
236.      957 FORMAT(1H0,10X,'IFFC CODE NOT EQUAL TO 10 ',//)
237.      AGRI=1./RVAR(2) ; HGRI=AGRI/2.
238.      IAX=LTKEY-89 ; IOX=LGKEY-180
239.      ALV=FL0AT(IAX) ; 0LV=FL0AT(IOX)

```

GIN00530

GIN00950

```

240.      CLAT=ALV+HGRI-(RVAR(3)-1)*AGRI
241.      CLONG=BLV+HGRI+(RVAR(4)-1)*AGRI
242.      RLAT=CLAT*DEGRA ; RLONG=CLONG*DEGRA
243.      VAR=RVAR(5) ; GOT0970
244.      956 VAR=RVAR(1) ; GOT0970
245.      97C ICNT=ICNT+1 ; ICND=ICND+1
246.      101 CONTINUE
247.      CALL STAT(I)
248.      710 CONTINUE
249.      CALL EVIL(IIOUT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0)
250.      IF (IBAD) 50, 53, 575
251.      53 IF (IREC1-2)600,70,600
252.      60C IF (IREC1-1)50,70,50
253.      575 IF (NEF - IFILE) 576, 577, 577
254.      576 NEF = NEF + 1
255.      IF (ISW(40)*EG.C)GOT0820
256.      READ(IDL,406)IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
257.      * (IDESC(K,NEF),K=1,17) ; IMDL=C
258.      82C CALL MOUNT(ITAPE,IDTIN(NEF))
259.      OUTPUT 'INPUT TAPE MOUNTED'
260.      WRITE (IIOUT,1413) IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
261.      1 (IDESC(K,NEF),K=1,17)
262.      OUTPUT '-----'
263.      REWIND ITAPE
264.      GO TO 50
265.      C END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
266.      577 IE0D=1 ; RETURN
267.      7C IF (ISW(12))73,73,71
268.      71 WRITE(IIOUT,72)KGDA,KGM0,KGYR,KGHM
269.      72 FORMAT('DATE=',3I3,15)
270.      C
271.      C -----
272.      C
273.      73 IF (ISW(29)*EG.C)GOT01730
274.      IF (ISW(29)*EG.2)GOT0170C
275.      C
276.      C ----- PROCESS ONLY SELECTED SOURCE CODES
277.      C
278.      D01650,=1,16
279.      IF (ISRC(J)*EG.C)GOT0850
280.      IF (IS0RC-ISRC(J))1650,1730,1650
281.      165C CONTINUE
282.      GOT0850
283.      C
284.      C ----- IGNORE SELECTED SOURCE CODES
285.      C
286.      170C D01710,=1,16
287.      IF (ISRC(J)*EG.C)GOT01730
288.      IF (IS0RC-ISRC(J))1710,50,1710
289.      171C CONTINUE
290.      C
291.      C -----
292.      C
293.      173C IF (ISW(60)*EG.1)GOT01731
294.      RLAT=DLAT*DEGRA
295.      RLONG=CLONG*DEGRA
296.      1731 IF (NX*EG.C)GOT0418
297.      C
298.      C CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF
299.      C THE 1967 INTERNATIONAL GRAVITY FORMULA

```

GIN01090  
GIN01100  
GIN01110  
GIN01120  
GIN01240  
GIN01250  
GIN01270  
GIN01280  
GIN01290  
GIN01300  
GIN01310  
GIN01390  
GIN01400  
GIN01410

```

300. C
301. C THIS SUB DECODES ONLY THAT VARIABLE TO BE PLOTTED.
302. C NX DETERMINES THE VARIABLE.
303. C
304. C IF(NX.EG.11)
305. 1      BUTPUT ' CANNOT PLOT OBSERVED GRAVITY'
306. 2      STOP
307. C
308. C
309. C TO AVOID PLOTTING INVALID DATA POINTS
310. C IF((NX.EG.6 .OR.
311. 1      NX.EG.7 .OR.
312. 2      NX.EG.9 .OR.
313. 3      NX.EG.10 .OR.
314. 4      NX.EG.14 .OR.
315. 5      NX.EG.15).AND.
316. 6      VAR.GT. 990.C)
317. 7      GO TO 50
318. C
319. C CORRECTION FOR FREE AIR OR BOUGUER FOR 67 G FORMULA
320. C RLAT = DLAT * DEGRA
321. C DG=3.2-(13.6*(SIN(ABS(RLAT))**2))
322. C
323. C IF((NX.EG.6 .OR.
324. 1      NX.EG.7 .OR.
325. 2      NX.EG.9 .OR.
326. 3      NX.EG.10 .OR.
327. 4      NX.EG.14 .OR.
328. 5      NX.EG.15).AND.
329. 6      IREC1.EG. 1)
330. 7      VAR = VAR + DG
331. C
332. C
333. C PLT(NX)=VAR
334. 109 CONTINUE
335. 418 CONTINUE
336. KGDA0=KGDA
337. KGM00=KGM0
338. KGYR0=KGYR
339. KGHM0=KGHM
340. IF(NX)1100,1200,1100
341. 1100 DATA=PLT(NX)
342. X      BUTPUT DATA
343. 1200 RETURN
344. C
345. C ----- GETG DLT INPUT LOGIC
346. C
347. 501 IF(IDLT.EG.1)GOTO514
348. IF(ICHA.EG.1)GOTO507
349. C
350. C ----- INPLT AN ELEMENT OF LIST OF DEGR. SQUAR. NEEDED
351. C
352. READ(IL1,502,END=550)LA1C,LATC,L010C,L0NC
353. 502 FORMAT(2(I2,I1))
354. C
355. 507 CONTINUE / ICHA=0
356. C
357. C ----- INPLT A MEMBER OF DLT
358. C
359. READ(ICL,503,END=540)IDEN,INBR,LA10T,LAT,L0101,L0N1,L0102,L0N2

```

GIN01640  
GIN01650  
GIN01660  
GIN01670  
GIN01680  
GIN01690

```

360.      503 FORMAT(A4,1X,I6,3(I2,I1))
361.      C
362.      C ----- TESTING DLT FOR CONSISTENCY
363.      C
364.      IF(L0101.NE.L0102)PRINTEC4 ; BUTPUT L0101,L0102 ; STOP
365.      504 FORMAT(1HC,10X,'ERROR IN DLT',/,2CX,'L0101 NOT EQUAL TO L0102',/)
366.      C
367.      IF(IDEN.EG.IBL)GOTO530
368.      IF(IDEN.EG.ITERI)GOTO540
369.      C
370.      L010=L0101 ; ILN=L0N1-L0N2+1
371.      C
372.      C ----- SIMPLY SCANNING OF DLT AND MATCHING LIST
373.      C
374.      505 KEY1=LA10T-LA10C ; KEY2=L010-L010C
375.      KEY3=LAT=LATC
376.      C
377.      D0506I=1,IL0
378.      L0N=L0N1-I+1
379.      KEY4=L0N-L0NC
380.      IF(KEY1)520,510,506
381.      510 IF(KEY2)520,511,506
382.      511 IF(KEY3)520,512,506
383.      512 IF(KEY4)520,513,506
384.      506 CONTINUE
385.      C
386.      C ----- SKIPPING UNNECESSARY RECORDS
387.      C
388.      D0509I=1,INBR
389.      CALL BLFIG(ITAPE,IEND,1,1,
390.      *      IBLK,32,KMAX,KN,IST,ILT,BLF,BLI,BLK,ICNT)
391.      IF(IEND.EG.1)GOTO508
392.      509 CONTINUE
393.      GOTO 507
394.      C
395.      C ----- ERROR CONDITIONS
396.      C
397.      508 PRINT534 ; IE0D=1 ; RETURN
398.      534 FORMAT(1HC,10X,'INCORRECT DLT - FOUND EOF WHILE SKIPPING REC.',/)
399.      C
400.      C ----- CONDITIONAL BRANCH TO READ PROCESS
401.      C
402.      513 ICND=0 ; IDLT=1 ; BUTPUT 'PROCESS',INBR ; GOTO52
403.      514 IF(ICND.EG.INBR)IDLT=0 ; GOTO501
404.      GOTO52
405.      C
406.      C ----- INPUT AN ELEM. OF MATCHING LIST
407.      C
408.      520 READ(ILI,502,END=550)LA10C,LATC,L010C,L0NC
409.      GOTO505
410.      C
411.      C ----- COMPLETION MESSAGE
412.      C
413.      550 PRINT551 ; IE0D=1 ; BUTPUT ICNT ; RETURN
414.      551 FORMAT(1HC,10X,'AREA PROCESSED - STOP',/)
415.      C
416.      C ----- CHANGE TAPE REEL
417.      C
418.      530 IDLT=0 ; ICHA=1 ; GOTO575
419.      C

```

```
420. C ----- END OF DLT REACHED
421. C
422. 540 PRINT 541 ; IEOD=1 ; RETURN
423. 541 FORMAT(1H0,10X,'END OF DLT REACHED - STOP',/)
424. END
```



AD-A035 454

WHOI-77-2  
GRAVITY DATA PROCESSING PROGRAMS.(U)

FEB 77

CARL BOWIN

UNCLASSIFIED WOODS HOLE OCEANOGRAPHIC INSTITUTION, MA

5 OF 6  
AD-A  
035 454



520	CC4C1	530	CC4E5	534	CC49A	540	CC4EA	541	CC4FO	550	CC4CC
551	CC4CB	575	CC2FF	576	CC302	577	CC35C	600	CC2FB	621	CC2A1
710	CC2EC	810	CC14C	811	CC117	820	CC325	900	CC064	912	CC084
913	CC075	920	CC15F	921	CC15F	922	CC1AA	923	CC1D4	924	CC177
950	CC237	953	CC231	954	CC298	955	CC2A5	956	CC2E4	957	CC2B1
970	CC2E7	1100	CC3F7	1200	CC3FA	1405	CC090	1410	CC0F9	1413	CC0137
1650	CC383	1700	CC389	1710	CC390	1730	CC394	1731	CC39F	2414	CC01DF
2415	CC1E4										

## LOCAL VARIABLES (3375 WORDS):

CC00C	GETGS	CC001	IGSU	CC025	IFMT	CC051	BUF	CC0691	BLI	CC0711	BLK
CC0791	IS	CC07F5	IL	CC0859	ILBN	CC086C	IPIE	CC0881	NVAR	CC0895	IVFM
CC08F9	RVAR	CC08FE	ICTIN	CC0912	IBK	CC0926	ITK	CC093A	IDENS	CC094E	IDESC
CC0AA2	ICTOT	CC0AB6	JBK	CC0ACA	JTK	CC0ADE	JDENS	CC0AF2	JDESC	CC0C46	PLY
CC0C5A	IA	CC0C7C	IAFMT	CC0C86	IASH	CC0CA5	ISRC	CC0CB9	ITERI	CC0CBA	IBL
CC0CBB	IFLAG	CC0CBC	IFIN	CC0CBD	IFLGI	CC0CBE	J	CC0CBF	IST	CC0CD3	ILT
CC0CE7	ILI	CC0CE8	IDL	CC0CES	ICND	CC0CEA	ICHA	CC0CEB	ICENT	CC0CEC	ITN
CC0CED	IIBUT	CC0CEE	KMAX	CC0CEF	KN	CC0CFC	DEGRA	CC0CF1	RADEG	CC0CF2	DLAB8
CC0CF3	CLAT8	CC0CF4	DL8LE	CC0CF5	DL8RI	CC0CF6	NZER8	CC0CF7	KGDA8	CC0CF8	KGM88
CC0CF9	KGYR8	CC0CFA	KGHM8	CC0CFB	NRECT	CC0CFC	IREC1	CC0CFD	INDICA	CC0CFE	IDLT
CC0CFF	NEF	CC0D00	IFILE	CC0D01	K	CC0D02	IMDL	CC0D03	I	CC0D04	IFMS1
CC0D05	IFMS2	CC0D06	IKSL	CC0D07	IST1	CC0D08	ILT1	CC0D09	IEND	CC0D0A	ICNT
CC0D0B	IS8RC	CC0D0C	CLAT	CC0D0D	DL8NG	CC0D0E	LTKEY	CC0D0F	LGKEY	CC0D10	VAR
CC0D11	A	CC0D12	AGRI	CC0D13	HGRI	CC0D14	IAX	CC0D15	I8X	CC0D16	ALV
CC0D17	8LV	CC0D18	CLAT	CC0D19	CL8NG	CC0D1A	IBAD	CC0D1B	DG	CC0D1C	LA10C
CC0D1C	LATC	CC0D1E	L81CC	CC0D1F	L8NC	CC0D2C	IDEN	CC0D21	INBR	CC0D22	LA10T
CC0D23	LAT	CC0D24	L81C1	CC0D25	L8N1	CC0D26	L8102	CC0D27	L8N2	CC0D28	L810
CC0D29	IL8	CC0D2A	KEY1	CC0D2B	KEY2	CC0D2C	KEY3	CC0D2D	L8N	CC0D2E	KEY4

## BLANK COMMON (C WORDS)

## ENTRY POINTS:

C0000 GETGS

## INTRINSIC SUBPROGRAMS USED:

ABS FLBAT SIN

## EXTERNAL SUBPROGRAMS REQUIRED:

BUFIC	ENC18	ENCLT
F:101	F:102	F:103
9BCDREAD	9BCDWRIT	9DEC8DE
9REWIND	9SETUPN	9SIN

EVIL	ISW	STAT
F:104	F:105	9BCDRDEE
9ENC18L	918DATA	9BCDRDEE
9ST8F		9IT8R

SETSKP	STAT
F:108	9BCDRDEE
9IT8R	9BCDRDEE

NUMBER OF X CARDS IGNORED: 1

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	1276	004FC
CONSTANTS:	20	00014
LOCAL VARIABLES:	3375	00D2F
TEMPS:	20	00014
	-----	-----
TOTAL PROGRAM:	4691	C1253

```

1.      SUBROUTINE GETH(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATZ,DATAW,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHP,IE0D)
3.      C
4.      C      VERSION OF 6 JUNE 1972,  ALLOWS FOR LETTERED STATION NUMBERS
5.      C      VERSION OF 22 MAR 1972,  IGNORES COL 1 AND CHECKS IF LAT
6.      C      AND LONG ARE ZERO
7.      C      VERSION OF 18 MAR 1972  --  FIRST GENERATION OF ROUTINE
8.      C
9.      C
10.     C  SUBROUTINE GETH,  FOR READING HEAT FLOW DATA
11.     C
12.     C
13.     DIMENSION PLT(8),M(6)
14.     400  IIN = 105
15.     IIBUT = 108
16.     IE0D=0
17.     KGDA=0
18.     KGM0=0
19.     410  CONTINUE
20.     READ(ITAPE,42) IN0,AN0,M,STAID,LAT,A1,ILATM,KNS,L0NG,A2,ILOM,KEW,
21.     1  IHEIT,GRAD,C0ND,HF,ICC,IREF,IYR
22.     42  FORMAT(1X,I4,A1,1X,6I1,1X,A8,1X,I2,A1,I2,A1,1X,I3,A1,I2,A1,1X,
23.     1  I5,1X,F4.2,1X,F4.2,1X,F5.2,3X,A1,I4,1X,I2)
24.     CALL STAT(I)
25.     CALL EVIL(IIBUT,I,IBAD,KGDA0,KGM00,KGYR0,KGHP0)
26.     IF(IBAD)410,53,900
27.     53  CONTINUE
28.     KGYR=IYR
29.     KGHP=IN0
30.     RLAT=ILATM
31.     RLOM=ILOM
32.     C
33.     C      CHECKING FOR ZERO LAT AND LONG
34.     C
35.     IF (LAT)70,60,70
36.     60  IF(ILATM)70,62,70
37.     62  IF(L0NG)70,64,70
38.     64  IF(ILOM)70,410,70
39.     70  CALL NAVIN(LAT,RLATM,KNS,L0NG,RLOM,KEW,RLAT,RL0NG)
40.     KGDA0=KGDA
41.     KGM00=KGM0
42.     KGYR0=KGYR
43.     KGHP0=KGHP
44.     PLT(1)=IN0
45.     PLT(2)=IHEIT
46.     PLT(3)=HF
47.     PLT(4)=GRAD
48.     PLT(5)=C0ND
49.     PLT(6)=M(1)*100000+M(2)*10000+M(3)*1000+M(4)*100+M(5)*10+M(6)
50.     PLT(7)=IREF
51.     PLT(8)=IYR
52.     IF(NX)80,85,80
53.     80  DATAX=PLT(NX)
54.     85  DATAY=PLT(NY)
55.     DATAZ=IHEIT
56.     DATAW=IN0
57.     RETURN
58.     900  IE0D=1
59.     RETURN

```

60•

END

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
ANG	R	SCALR	00012 V	1	A1	R	SCALR	00015 V	1	A2	R	SCALR	00019 V	1
COND	R	SCALR	0001E V	1	DATAM	R	SCALR	*00033 V	DUMMY	DATAX	R	SCALR	*00030 V	DUMMY
DATAY	R	SCALR	*00031 V	DUMMY	DATZ	R	SCALR	*00032 V	DUMMY	EVIL	R	SPR8G	EXTERN	1
GETH	R	SCALR	00000 V	1	GETH	R	SPR8G	00000 P	1	GRAD	R	SCALR	0001D V	1
HF	R	SCALR	0000F V	1	I	I	SCALR	00003A V	DUMMY	IBAD	I	SCALR	00024 V	1
ICD	I	SCALR	00020 V	1	IE8D	I	SCALR	00010 V	1	IEIT	I	SCALR	0001C V	1
IIN	I	SCALR	0000F V	1	IE8T	I	SCALR	00011 V	1	ILATH	I	SCALR	00016 V	1
IL8M	I	SCALR	0001A V	1	IN8	I	SCALR	00011 V	1	IREF	I	SCALR	00021 V	1
ITAPE	I	SCALR	*00028 V	DUMMY	IYR	I	SCALR	00022 V	1	KEW	I	SCALR	00018 V	1
KGDA	I	SCALR	*00036 V	DUMMY	KGDA8	I	SCALR	00025 V	1	KGHM	I	SCALR	*00039 V	DUMMY
KGHM8	I	SCALR	00028 V	1	KGP8	I	SCALR	*00037 V	DUMMY	KGM88	I	SCALR	00026 V	1
KGYR	I	SCALR	*00038 V	DUMMY	KGYR8	I	SCALR	00027 V	1	KNS	I	SCALR	00017 V	1
LAT	I	SCALR	00014 V	1	LBNG	I	SCALR	00018 V	1	M	I	ARRAY	00009 V	6
NAVIN	I	SPR8G	EXTERN	1	NW	I	UNUSED*	0002F V	DUMMY	NX	I	SCALR	*0002C V	DUMMY
NY	I	SCALR	*0002D V	DUMMY	NZ	I	UNUSED*	0002E V	DUMMY	PLT	R	ARRAY	00001 V	8
RLAT	R	SCALR	*00034 V	DUMMY	RLATH	R	SCALR	00029 V	1	RL8M	R	SCALR	0002A V	1
RL8NG	R	SCALR	*00035 V	DUMMY	STAD	R	SCALR	00013 V	1	STAT	R	SPR8G	EXTERN	1

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
42	00037	60	0006C	62	0006E	64	00070
80	000AD	400	00013	410	0001D	900	0008A

## LOCAL VARIABLES (43 WORDS):

00000	GETH	00009	M	0000F	IIN	00011	IN8
00012	AN8	00014	LAT	00015	A1	00016	KN8
00018	LBNG	0001A	IL8M	0001B	KEW	0001C	GRAD
0001E	CBND	0002C	ICD	00021	IREF	00022	IYR
00024	IBAD	00026	KGM88	00027	KGYR8	00028	KGHM8
0002A	RL8M					00029	RLATH

BLANK COMMON (0 WORDS)

ENTRY PRINTS:

COCCC GETH

## EXTERNAL SUBPROGRAMS REQUIRED:

EVIL	NAVIN	STAT	98CDREAD	916DATA
916LUSA	SIT8R	9SETUPN		

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	189	000BD
CONSTANTS:	0	0000C
LOCAL VARIABLES:	43	0002B
TEMPS:	21	00015
TOTAL PROGRAM:	253	000FD

```

1.      SUBROUTINE GETL(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAX,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHM,IE0D)
3.      C      VERSION 16 APRIL 1975, TO CORRECT KEYPUNCH ERROR
4.      C      VERSION OF 8 APRIL 1975, TO ADD SELECTION OF TOTAL
5.      C      ACCELERATION WITH SIGN OF RADIAL COMPONENT
6.      C      VERSION OF 23 MAR 1973, ADDING BOUGUER CALCULATION
7.      C      VERSION OF 28 FEB 1973, CHANGE FROM MSC FORMAT TO WHOI FMT
8.      C      VERSION OF 27 OCT 1972
9.      C      SUBROUTINE GETL, READS LUNAR DATA FROM MSC SFSN PROGRAM
10.     C
11.     C      SSW(34) = 1, TO READ LUNAR DATA ON 2 CARDS
12.     C      SSW(35) = 1, TO WRITE LUNAR DATA ON 2 CARDS
13.     C
14.     C      DIMENSION PLT(15)
15.     C      DIMENSION ID(7)
16.     C      DATA ISRT/C/
17.     C      IF(ISRT.NE.C) GO TO 50
18.     C      IIN=105
19.     C      II0LT=108
20.     C      IE0D=C
21.     C      OUTPUT ' SUBROUTINE GETL, VERSION OF 15 APRIL 1975'
22.     C      KGDA0=0
23.     C      KGM00=0
24.     C      KGYR0=0
25.     C      KGHM0=0
26.     C      DEGRA=1.745329E-2
27.     C      ISRT = 1
28.     C      *****
29.     C
30.     C      SETTING DENSITY VALUES FOR RIM AND CRUST
31.     C
32.     C      RIMD=2.50
33.     C      CRUSTD=2.73
34.     C      *****
35.     C
36.     C      END OF INITIALIZATION
37.     C
38.     50  IE0D=C
39.     C      IF(ISW(34)=1)51,55,51
40.     51  READ(ITAPE,100) IREC,IS0RC,KDA,KM0,KYR,KHM,SEC,DLAT,DLONG,SVEC,
41.     C      * ALTL,AZ,SINC,STAC,SNAC,FA,THE0R,KSS1,KSSN,KSSR,KSSA,ELEV,ELFL,
42.     C      * ID,LTKEY,LGKEY
43.     100 FORMAT(I1,I4,3I2,I4,F5.2,2F9.4,F8.3,F7.3,F6.2,F6.2,F6.1,F6.1,
44.     C      * F6.1,F9.2,4I3,2F7.3,7A1,1X,2I3)
45.     C      GO TO 58
46.     55  READ(ITAPE,102) IREC,IS0RC,KDA,KM0,KYR,KHM,SEC,DLAT,DLONG,SVEC,
47.     C      * ALTL,AZ,SINC,STAC,SNAC,FA,THE0R,KSS1,KSSN,KSSR,KSSA,ELEV,ELFL,
48.     C      * ID,LTKEY,LGKEY
49.     102 FORMAT(I1,I4,3I2,I4,F5.2,2F9.4,F8.3,F7.3,F6.2,F6.2,F6.1,F6.1/
50.     C      * F6.1,F9.2,4I3,2F7.3,7A1,1X,2I3)
51.     C
52.     58  CALL STAT(I)
53.     C      CALL EVIL(II0LT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM)
54.     C      IF(IBAD)50,60,900
55.     60  CONTINUE
56.     C      IF(IREC.EQ.1) GO TO 65
57.     C      OUTPUT 'IREC DOES NOT = 1'
58.     C      IF(ISW(34).NE.1) GO TO 50
59.     C      FOR DATA READ FROM 2 CARDS
61  READ(ITAPE,62)

```

```

60.      62  FORMAT(1X)
61.      65  GO TO 50
62.      65  CONTINUE
63.      KGDAB= KDA
64.      KGM00= KM0
65.      KGYR0= KYR
66.      KGHM = KHM
67.      RLAT = DLAT*DEGRA
68.      RL0NG= DL0NG*DEGRA
69.      G0BS = FA+THE0R
70.      PLT(1)= SVEC
71.      PLT(2)= SVEC -1738.0
72.      IF(NX.NE.3.0R.NX.NE.4) GO TO 80
73.      IF (ALTL.LT.0.001)PLT(3)=0.; PLT(4)=0; GO TO 50
74.      80  PLT(3) = ALTL
75.      PLT(4)=(SVEC-ALTL)-1738.0
76.      85  PLT(5)=AZ
77.      PLT(6)=SINC
78.      PLT(7)=STAC
79.      PLT(8)=SNAC
80.      PLT(9)=FA
81.      PLT(10)=THE0R
82.      PLT(11) = G0BS
83.      PLT(12) = ELEV
84.      IF(NX.EQ.13.AND.ALTL.LT.0.001) PLT(13)=0.; GO TO 50
85.      PLT(13)=ELFL
86.      IF(ELFL)22,23,23
87.      C   SETTING DENSITY TO THAT OF CRATER RIM
88.      22  DENSC=RIMC
89.      GO TO 24
90.      C   SETTING DENSITY TO THAT OF CRUST
91.      23  DENSC=CRUSTD
92.      24  CONTINUE
93.      BG=FA = ((DENSC*ELFL)*0.04185)
94.      PLT(14)=BG
95.      C   DETERMINING TOTAL ACCELERATION MAGNITUDE
96.      CALL TOTAC(STAC,SNAC,FA,TACEL)
97.      PLT(15)= TACEL
98.      IF(NX)90,95,9C
99.      9C  DATA=PLT(NX)
100.     95  DATAY=PLT(NY)
101.     DATAZ= PLT(2)
102.     RETURN
103.     90C IE0D = 1
104.     RETURN
105.     END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
ALTL	R	SCALR	00028	1	AZ	R	SCALR	0002C	1	BG	R	SCALR	0003E	1
CRUSTD	R	SCALR	00020	1	DATAH	R	UNUSED	00048	1	DATAH	R	SCALR	00045	1
DATAY	R	SCALR	00046	1	DATZ	R	SCALR	00047	1	DEGRA	R	SCALR	0004E	1
DENSC	R	SCALR	0003D	1	ELFL	R	SCALR	00038	1	DEGRA	R	SCALR	00029	1
ELEV	R	SCALR	00036	1	GETL	R	SCALR	00037	1	EVIL	R	SPRGG	EXTERN	1
FA	R	SCALR	00030	1	I	R	SPRGG	00000	1	GETL	R	SCALR	00000	1
GBBS	R	SCALR	0003C	1	I	R	SCALR	0003A	1	IBAD	R	SCALR	0003B	1
ID	R	SCALR	00010	1	I	R	SCALR	0004F	1	I'N	R	SCALR	00018	1
IIBLT	R	SCALR	00019	1	I	R	SCALR	00021	1	ISRC	R	SCALR	00022	1
ISRT	R	SCALR	00017	1	I	R	SPRGG	EXTERN	1	ITAPE	R	SCALR	00040	1
KCA	R	SCALR	00023	1	I	R	UNUSED	0004B	1	KGDAB	R	SCALR	0001A	1
KGMH	R	SCALR	00018	1	I	R	SCALR	0001D	1	KGMH	R	UNUSED	0004C	1
KHM	R	SCALR	00026	1	I	R	SCALR	0004D	1	KGYR	R	SCALR	0001C	1
KSSN	R	SCALR	00033	1	I	R	SCALR	00024	1	KSSA	R	SCALR	00035	1
KYR	R	SCALR	00025	1	I	R	SCALR	00034	1	KSST	R	SCALR	00032	1
NH	R	UNUSED	00044	1	I	R	SCALR	00039	1	LTKEY	R	SCALR	00038	1
AZ	R	UNUSED	00043	1	I	R	SCALR	00041	1	NY	R	SCALR	00042	1
RLAT	R	SCALR	00049	1	I	R	ARRAY	00041	1	RIMD	R	SCALR	0001F	1
SINC	R	SCALR	0002D	1	I	R	SCALR	0004A	1	RIMD	R	SCALR	00027	1
STAT	R	SPRGG	EXTERN	1	I	R	SCALR	0002F	1	STAC	R	SCALR	0002E	1
THEOR	R	SCALR	00031	1	I	R	SCALR	0002A	1	TACEL	R	SCALR	0003F	1
					I	R	SPRGG	EXTERN	1					

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
22	00122	24	00127	50	0003B	51	00042
58	00033	61	00044	62	00008	65	000DB
85	00105	95	0013C	100	00063	102	0009C

## LOCAL VARIABLES (64 WORDS):

COCCC	GETL	00001	PLT
CO01A	KGDAB	0001B	KGMH
CO02C	CRUSTD	00021	IREC
CO026	KHM	00027	SEC
CO02C	AZ	0002D	SINC
CO032	KSST	00033	KSSN
CO038	LTKEY	00039	LGKEY
CO03E	BG	0003F	TACEL

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

COCCC GETL

## EXTERNAL SUBPROGRAMS REQUIRED:

EVIL	ISN	STAT	F:103	F:105	F:108
SECCREAD	SENCIOL	918DATA	SSSETUPN		

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	325	00145
CONSTANTS:	7	00007
LOCAL VARIABLES:	64	00040
TEMPS:	17	00011
TOTAL PROGRAM:	413	0019D

```

1.      SUBROUTINE GETM(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAW,RLAT,
2.      1RLONG,KGDA,KGM0,KGYR,KGHM,IE00)
3.      C
4.      C      SUBROUTINE GETM,FOR READING BATHYMETRY AT MBATR FORMAT
5.      C
6.      C      CHANGED 8 NOV. 1971 BY R.C. GROMAN TO UNIFY DEFINITION OF
7.      C      FLT(1)=TIME IN ALL 'GET' SUBROUTINES
8.      C
9.      DIMENSION FLT(7)
10.     IE00=0
11.     II0UT=108
12.     14 CALL ENDI0
13.     15 READ(ITAPE,16)KGDA,KGM0,KGYR,KGHM,ITZ,DATA,MTAB,CDFM,CDM,DLAT,DL0N
14.     1G,DIS,DIR,SPD
15.     16 FORMAT(3I2,I4,1X,I3,1X,F5.0,1X,I2,2X,F5.0,2X,F5.0,1X,F7.3,1X,F8.3,
16.     11X,F7.1,1X,F3.0,1X,F4.1)
17.     CALL STAT(I)
18.     CALL EVIL(II0UT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0)
19.     IF(IBAD)14,30,65
20.     65 IE00=1
21.     RETURN
22.     30 RLAT=DLAT*(1.0/57.29578)
23.     RL0NG=DL0NG*(1.0/57.29578)
24.     FLT(1)=KGHM
25.     FLT(2)=CDFM
26.     FLT(3)=CDM
27.     FLT(4)=DIS
28.     FLT(5)=DIR
29.     FLT(6)=SPD
30.     FLT(7)=DATA
31.     KGDA0=KGDA
32.     KGM00=KGM0
33.     KGYR0=KGYR
34.     KGHM0=KGHM
35.     C      SELECTING DATA POINT TO BE PLOTTED
36.     IF(NX)110,120,110
37.     110 DATAX=FLT(NX)
38.     120 DATAY=FLT(NY)
39.     DATAZ=FLT(NZ)
40.     DATAW=FLT(NW)
41.     RETURN
42.     END

```

NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS
CDPM	R	SCALR	0000C V	1	CDM	R	SCALR	0000C V	1	DATA	R	SCALR	0000A V	1
DATA*	R	SCALR	00021 V DUMMY	1	DATA*	R	SCALR	0001E V DUMMY	1	DATA*	R	SCALR	0001F V DUMMY	1
DATAZ	R	SCALR	00020 V DUMMY	1	DIR	R	SCALR	00011 V	1	DIS	R	SCALR	00010 V	1
CLAT	R	SCALR	0000E V	1	DLNG	R	SCALR	000CF V	1	ENDI0	R	SPR0G	00010 V	1
EVIL	R	SPR0G	0000E V	1	GETM	R	SCALR	0000C V	1	GETM	R	SPR0G	00000 P	1
IIBUT	I	SCALR	00013 V	1	IBAC	I	SCALR	00014 V	1	IE0D	I	SCALR	00028 V DUMMY	1
KGDA	I	SCALR	00008 V	1	ITAPE	I	SCALR	00019 V DUMMY	1	ITZ	I	SCALR	00009 V DUMMY	1
KGPM8	I	SCALR	00018 V	1	KGDA8	I	SCALR	00015 V	1	KGHM	I	SCALR	00027 V DUMMY	1
KGPM8	I	SCALR	00018 V	1	KGPM8	I	SCALR	00025 V DUMMY	1	KGPM8	I	SCALR	00016 V	1
KGPM8	I	SCALR	00018 V	1	KGPM8	I	SCALR	00025 V DUMMY	1	MTAB	I	SCALR	0000B V	1
NW	I	SCALR	00026 V DUMMY	1	NX	I	SCALR	00017 V	1	NY	I	SCALR	0001B V DUMMY	1
NZ	I	SCALR	0001D V DUMMY	1	PLT	R	ARRAY	000C1 V	7	RLAT	R	SCALR	00022 V DUMMY	1
RLONG	R	SCALR	0001C V DUMMY	1	SPC	R	SCALR	00012 V	1	STAT	R	SPR0G	00022 V DUMMY	1

NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS
14	CC017		15	00019	16	CC02B		30	00053
120	CC077							65	00050

## LOCAL VARIABLES (25 WORDS):

0000C GETM  
0000C CDFM  
00012 SPD  
00018 KGPM8

## BLANK COMMON (0 WORDS)

## ENTRY PRINTS:

0000C GETM

## EXTERNAL SUBPROGRAMS REQUIRED:

ENDI0 SIT0R	EVIL SSETUPN	STAT	F:101	F:103	F:105	9BCDREAD	910DATA

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
129	CCC81
2	CCC02
25	CCC19
17	CCC11
173	CCCAC

TOTAL PROGRAM: 173

```

1.      SLBRoutine GETP(ITAPE,NX,NY,NZ,NW,DATA1,DATA2,DATA3,DATA4,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHP,IE0D)
3.      C  SLBRoutine GETP,  FOR INPUT OF SEISMIC REFRACTION
4.      C  PROFILE DATA AT SPFMT FORMAT
5.      C
6.      C
7.      C  SSW(32)  LP TO READ SPFMT DATA ON TWO CARDS
8.      C  SSW(33)  LP TO WRITE SPFMT DATA ON TWO CARDS
9.      C
10.     DIMENSION FLT(15) ,IDESC(6)
11.     DATA K9,NEW,NNS/'9  ','W  ','S  ','/
12.     C
13.     C  USES SLBRoutines EVIL, ISW, STAT ,DMTOR
14.     C
15.     C  *****
16.     C  *****
17.     C
18.     IIN = 105
19.     II0UT = 108
20.     KGDA=C
21.     KGM0=C
22.     KGYR=C
23.     C
24.     C  READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
25.     10  CONTINUE
26.     IF (ISW(32)) 15,12,15
27.     12  READ(ITAPE,990)IREC1,ISTA,KEY,LAT,LATM,KNS,LONG,L0M,
28.     1  KEW,J1,K1,J2,K2,J3,K3,J4,K4,J5,K5,J6,K6,J7,K7,J8,K8,
29.     2  IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,STHIK,CRVN,
30.     3  WGTN,AVWTN,CRVW,WGTW,AVWTW
31.     990  FORMAT(I1,I4,A1,I2,I2,A1,I3,I2,A1,8(I2,I3),I2,I4,4I1,
32.     1  I1,I2,6A2,2F4.1,F3.1,2F6.0,1X,F3.1,2F6.0,5X)
33.     GO TO 18
34.     15  READ(ITAPE,991)IREC1,ISTA,KEY,LAT,LATM,KNS,LONG,L0M,
35.     1  KEW,J1,K1,J2,K2,J3,K3,J4,K4,J5,K5,J6,K6,J7,K7,J8,K8,
36.     2  IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,STHIK,CRVN,
37.     3  WGTN,AVWTN,CRVW,WGTW,AVWTW
38.     991  FORMAT(I1,I4,A1,I2,I2,A1,I3,I2,A1,8(I2,I3),I2,I4,4I1,
39.     1  I1,I2,10X,6A2,2F4.1,F3.1,2F6.0,1X,F3.1,2F6.0,5X)
40.     18  CALL STAT(I)
41.     CALL EVIL(II0UT,I,IBAD,KDA,KM0,KYR,ISTA0)
42.     IF (IBAD) 10, 20, 900
43.     C  CHECKING IF KEY = 9
44.     19  IF (KEY=K9)20,10,20
45.     20  ELEV=NELEV
46.     ELEV=ELEV * 0.01
47.     VMANT=IMANT
48.     VMANT=VMANT * 0.1
49.     KGHP=ISTA
50.     50  IF (N1=2) 70,60,70
51.     C  SEA SEISMIC PROFILE
52.     60  VELW= 1.5
53.     HEIGT=-ELEV
54.     GO TO 80
55.     C  LAND SEISMIC PROFILE
56.     70  VELW= 0.0
57.     HEIGT=ELEV
58.     C  MAIN PLOTTING LOOP
59.     80  RLATM = LATM

```

```

60.      RLBM = LBM
61.      RLAT = DMTOR(LAT,RLATM)
62.      RLONG = DMTOR(LONG,RLBM)
63.      IF(KNS=NNS)54, 52, 54
64.      52  RLAT = -RLAT
65.      54  IF(KEW=NEW)58,56,58
66.      56  RLONG = -RLONG
67.      58  PLT(1)=ISTA
68.      PLT(2)=HEIGT
69.      PLT(3)=VMANT
70.      PLT(4)=DINE
71.      PLT(5)=STH1K
72.      PLT(6)=CRVN
73.      PLT(7)=WGTH
74.      PLT(8)=AVWTN
75.      PLT(9)=CRVW
76.      PLT(10)=WGTH
77.      PLT(11)=AVWTW
78.      C SELECTING DATA TO BE PLOTTED
79.      IF(NX)110,120,110
80.      11C DATA=PLT(NX)
81.      12C DATAY=PLT(NY)
82.      DATAZ=PLT(NZ)
83.      DATAW=PLT(NW)
84.      IEOD=C
85.      RETURN
86.      90C IEOD=1
87.      RETURN
88.      END

```

NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS
AVMTN	R	SCALR	0004C V	1	AVMTN	R	SCALR	00043 V	1	CRVN	R	SCALR	0003E V	1
CRVN	R	SCALR	00041 V	1	DATAH	R	SCALR	*00058 V	DUMMY	DINE	R	SCALR	*00055 V	DUMMY
DATAY	R	SCALR	*00056 V	DUMMY	DATZ	R	SCALR	*00057 V	DUMMY	EVL	R	SCALR	0003C V	1
CHTBR	R	SPRGG	EXTRN		ELEV	R	SCALR	0004A V	1	HEIGT	R	SCALR	EXTRN	
GETP	I	SPRGG	0000C P		GETP	R	SCALR	0000C V	1	IDESC	I	ARRAY	00010 V	6
I	I	SCALR	00044 V	1	IBAC	I	SCALR	00045 V	1	IBUT	I	SCALR	0001A V	1
IE6C	I	SCALR	*0005F V	DUMMY	IN	I	SCALR	00019 V	1	ISTA	I	SCALR	0001C V	1
IMANT	I	SCALR	00034 V	1	IREC1	I	SCALR	00018 V	1	ITAPE	I	SCALR	*00050 V	DUMMY
ISTAB	I	SCALR	00049 V	1	ISN	I	SPRGG	EXTRN		J2	I	SCALR	00026 V	1
IYR	I	SCALR	00039 V	1	J1	I	SCALR	00024 V	1	J5	I	SCALR	0002C V	1
J3	I	SCALR	00028 V	1	J4	I	SCALR	0002A V	1	J8	I	SCALR	00032 V	1
V6	I	SCALR	0002E V	1	J7	I	SCALR	00030 V	1	KEY	I	SCALR	0001D V	1
KDA	I	SCALR	00046 V	1	KEH	I	SCALR	00023 V	DUMMY	KGM9	I	SCALR	*0005C V	DUMMY
KGDA	I	SCALR	*0005B V	DUMMY	KGMH	I	SCALR	*0005E V	DUMMY	KNS	I	SCALR	00020 V	1
KGYR	I	SCALR	*0005D V	DUMMY	KH9	I	SCALR	00047 V	1	K2	I	SCALR	00027 V	1
KYR	I	SCALR	00048 V	1	K1	I	SCALR	00025 V	1	K5	I	SCALR	0002D V	1
K3	I	SCALR	00029 V	1	K4	I	SCALR	00031 V	1	K8	I	SCALR	00033 V	1
K6	I	SCALR	0002F V	1	K7	I	SCALR	0003E V	1	LATH	I	SCALR	0001F V	1
K9	I	SCALR	00016 V	1	LAT	I	SCALR	0001E V	1	MET	I	SCALR	0003A V	1
LBM	I	SCALR	00022 V	1	LONG	I	SCALR	00017 V	1	NNS	I	SCALR	00018 V	1
NELEV	I	SCALR	00035 V	1	NEW	I	SCALR	*00051 V	DUMMY	NY	I	SCALR	*00052 V	DUMMY
NH	I	SCALR	*00054 V	DUMMY	NX	I	SCALR	00036 V	1	N2	I	SCALR	00037 V	1
NZ	I	SCALR	*00053 V	DUMMY	N1	I	SCALR	00039 V	1	PLT	R	ARRAY	00001 V	15
N3	I	SCALR	00038 V	1	N4	I	SCALR	0004E V	1	RLM	R	SCALR	0000F V	1
RLAT	R	SCALR	*00059 V	DUMMY	RLATH	R	SCALR	EXTRN		SHK	R	SCALR	0003D V	1
RLONG	R	SCALR	*0005A V	DUMMY	STAT	R	SPRGG	EXTRN		WGTH	R	SCALR	0003F V	1
VELW	R	SCALR	0004C V	1	VMANT	R	SCALR	0004B V	1					
WGTH	R	SCALR	00042 V	1										

LABEL	HEX L6C	LABEL	HEX L6C	LABEL	HEX L6C	LABEL	HEX L6C
10	0001D	18	000B6	19	000C6	20	000C9
50	000D7	56	000FD	58	00100	60	000DA
70	000EC	120	0011C	900	00128	990	00053
991	0009D						

## LOCAL VARIABLES (80 WORDS):

CCCC GETP	000C1 FLT	00018 NNS
CC019 IIN	0001A IIBUT	0001E LAT
CC01F LATM	00020 KNS	00024 J1
CC025 K1	00026 J2	0002A J4
CC02B K4	0002C J5	00030 J7
CC031 K7	00032 J8	00036 N1
CC037 N2	00038 N3	0003C DINE
CC03C STH-K	0003E CRVN	00042 WGTN
CC043 AVMTN	00044 I	00048 KYR
CC049 ISTAB	0004A ELEV	0004E RLAT
CC04F RLBM		

BLANK COMMENTS (0 WORDS)

## ENTRY POINTS:

C0000 GETP

## EXTERNAL SUBPROGRAMS REQUIRED:

DMT0R	EVIL	ISW	STAT	F:101	F:103	F:105	9BCDRE'D
910DATA	910LUSA	91T0R	9SETUPN				

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	299	C012B
CONSTANTS:	4	C0004
LOCAL VARIABLES:	80	C005C
TEMPS:	17	C0011
TOTAL PROGRAM:	400	C019C

```

1.      SUBROUTINE GETS(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAW,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHM,IEBC)
3.      C      VERSION 28 MAY 1974, C. DEAN, FOR 67 GRAVITY FORMULA
4.      C      VERSION OF 22 JAN 1972, CORRECTING RETURN INDICATION OF EOF
5.      C      VERSION OF 19 JAN 1972, ADDING UNCORRECTED DEPTHS TO SELECTION VARIABLES
6.      C
7.      C  SUBROUTINE GETS, FOR READING SEAG1 FORMAT
8.      C
9.      C      DIMENSION FLT(20)
10.     DATA IFLAG/C/
11.     IF (IFLAG.NE.C) GO TO 50
12.     IFLAG = 1
13.     C
14.     C      SSW(3) UP TO ADD CURRENT VELOCITIES TO SHIP S VELOCITIES
15.     C      SSW(12) UP TO LIST DATE IDENTIFICATION
16.     C
17.     C      USES SUBROUTINES ENDIO(DUMMY), SHTV, EVIL, STAT
18.     C      ASSUME STAT INITIALIZED IN MAIN PROGRAM
19.     C
20.     OUTPUT 1 GETS VERSION 28 MAY 1974 FOR 67 GRAVITY FORMULA
21.     10  II0LT = 108
22.     IEBC=C
23.     50  CALL ENDIO
24.     52  READ(ITAPE,12)IREC1,KGDA,KGM0,KGYR,KGHM,IDIF,
25.     1  RLAT,RLONG,KVN,KVE,K977,I0GR,KFA,KBG,KCVN,
26.     2  KCVE,KCDM,MTDC,MT,MAG1,MAG2,KETVB
27.     12  FORMAT(I1,3I2,I4,I3,2F5.6,2I5,I3,I4,5I5,
28.     1  I3,I2,I1,I4,I5)
29.     CALL STAT(I)
30.     CALL EVIL(II0LT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0)
31.     IF (IBAD)50,53,65
32.     C  CONVERTING TO FLOATING POINT
33.     53  XKCDM = KCDM
34.     XKFA=FLOAT(KFA)*0.1
35.     XKBG=FLOAT(KBG)*0.1
36.     VN=FLOAT(KVN)*0.01
37.     VE=FLOAT(KVE)*0.01
38.     IF (IREC1-1) 55,56,55
39.     55  IF (IREC1-2) 60,70,60
40.     C
41.     C      CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF
42.     C      THE 1967 INTERNATIONAL GRAVITY FORMULA
43.     C      AND NEW GEODETIC REFERENCE SYSTEM.
44.     C
45.     56  CONTINUE
46.     I0GR = I0GR - 14.0
47.     DG = 3.2-(13.6*(SIN(ABS(RLAT))**2))
48.     IF (XKFA-990.0) 57,58,58
49.     57  XKFA = XKFA + DG
50.     58  IF (XKBG-990.0) 59,70,70
51.     59  XKBG = XKBG + DG
52.     GO TO 70
53.     60  IF (IREC1-9)50,62,50
54.     62  READ(ITAPE,64)IREC9,I2,I21,ITEST
55.     64  FORMAT(I1,2I3,I4)
56.     IF (ITEST-6563)580,565,580
57.     565  WRITE(II0LT,570)
58.     570  FORMAT('E0R')
59.     GO TO 50

```

```

60.      58C  IF(IITEST=6665)68,65,68
61.      65  IEUD=1
62.      RETLRN
63.      68  WRITE(IIOUT,69)
64.      69  FORMAT('IREC1=9')
65.      GO TO 50
66.      70  IF(ISW(12))73,73,71
67.      71  WRITE(IIOUT,72)KGDA,KGM0,KGYR,KGHM
68.      72  FORMAT('DATE=',3I3,15)
69.      73  CONTINUE
70.      C LSE CURRENT VELOCITIES
71.      IF(ISW(3))80,80,75
72.      75  VN=VN+(FL0AT(KCVN)*0.1)
73.      VE=VE+(FL0AT(KCVE)*0.1)
74.      80  KK=0
75.      CALL SFTV(VN,VE,SPEED,XHEAD,KK)
76.      E0TV=FL0AT(KETV0)*0.1
77.      TMAG=(MAG1*10000)+MAG2
78.      KMAG2=(MAG2/1000)*1000
79.      XMAG=MAG2-KMAG2
80.      XREG=0.0
81.      XKRES=0.0
82.      XLDM=KCDM-MTDC
83.      XLDF=XLDM*0.54681
84.      PLT(1)=KGHM
85.      PLT(2)=XKCDM
86.      PLT(3)=XKFA
87.      PLT(4)=XKBG
88.      PLT(5)=SPEED
89.      PLT(6)=XHEAD
90.      PLT(7)=E0TV
91.      PLT(8)=MT
92.      PLT(9)=XMAG
93.      PLT(10)=XREG
94.      PLT(11)=XKRES
95.      PLT(12)=-SPEED
96.      PLT(13)=-XKCDM
97.      PLT(14)=-E0TV
98.      PLT(15)=-XKFA
99.      PLT(16)=-XHEAD
100.     PLT(17)=TMAG
101.     PLT(18)=XLDM
102.     PLT(19)=XLDF
103.     KGDA0=KGDA
104.     KGM00=KGM0
105.     KGYR0=KGYR
106.     KGHM0=KGHM
107.     C CHECKING FOR INVALID VALUES
108.     IF(NX=2)105,607,606
109.     606  IF(NX=13)605,607,605
110.     607  IF(KCDM)105,10,105
111.     605  IF(NX=2)105,610,615
112.     610  IF(KFA-9980)105,10,10
113.     615  IF(NX=4)105,620,640
114.     620  IF(KBG-9980)105,10,10
115.     640  IF(NX=17)105,650,105
116.     650  IF(MAG1)10,10,105
117.     C SELECTING DATA TO BE PLOTTED
118.     105  IF(NX)110,120,110
119.     110  DATAX=PLT(NX)

```

```
120.      120  DATAY=FLT(NY)
121.          DATAZ=FLT(NZ)
122.          DATAW=FLT(NW)
123.      X    0LTFUT DATAZ
124.          RETURN
125.          END
```

[illegible]

LOCAL VARIABLES (66 WORDS):

CCCC	GET5	CCCC1	FLT	CCCC15	IFLAG	CCCC16	IBUT	CCCC17	IREC1	CCCC18	IDIF
CCCC1	KVE	CCCC14	KVE	CCCC1B	K97Y	CCCC1C	IBGR	CCCC1D	MTA	CCCC1E	KBG
CCCC2	KVE	CCCC20	KVE	CCCC21	KCDY	CCCC2C	MTDC	CCCC2D	KFA	CCCC2E	MAG1
CCCC3	MAG2	CCCC26	KET	CCCC27	I	CCCC3B	IBAD	CCCC3D	KGDAG	CCCC3E	KMG00
CCCC4	KGR0	CCCC2C	KMG0	CCCC3C	KXCD	CCCC3E	KXFA	CCCC3F	KXBG	CCCC3G	KV
CCCC5	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I
CCCC6	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I
CCCC7	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I
CCCC8	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I
CCCC9	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I
CCCCA	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I
CCCCB	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I
CCCCC	VE	CCCC32	CG	CCCC3C	I	CCCC4C	I	CCCC3E	I	CCCC3F	I

BLANK COMMON (C WERCS)

ENTRY POINTS:

COCC GETS

## INTRINSIC SUBPROGRAMS USED:

ABS          FLOAT          SIN

## EXTERNAL SUBPROGRAMS REQUIRED:

END10	EVIL	ISH	SHTV	STAT	F:101	F:102	F:103
F:104	F:105	F:106	F:108	9BCDREAD	9BCDWRT	9END10L	910DATA
9IT0R	9PRINT	9RT0I	9SETUPN	9SIN			

NUMBER OF X CARDS IGNORED: 1

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	363	CC16B
CONSTANTS:	10	CC00A
LOCAL VARIABLES:	66	CC042
TEMPS:	17	CC011
TOTAL PROGRAM:	456	CC1C8

```
1.      SLRBTINE GETST(ITAPE,NX,NY,NZ,NW,DATAZ,CATAY,DATAZ,DATAW,  
2.      1 RLAT,RLONG,UDA,UMB,UYR,UMM,IEOD)  
3.      C  
4.      C      VERSION OF 30 JUNE 1971  
5.      C      DUMMY ROUTINE WHILE AWAITING A WORKING VERSION  
6.      C  
7.      RETURN  
8.      END
```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
DATAW	----	UNUSEDC00C9	V	DUMMY	DATAW	----	UNUSEDC00C6	V	DUMMY	DATAW	----	UNUSEDC0007	V	DUMMY
DATAZ	----	UNUSEDC00C8	V	DUMMY	GETST	----	SPRGG C00CC	P	DUMMY	GETST	----	UNUSEDC000C	V	DUMMY
IE8C	----	UNUSEDC00C10	V	DUMMY	ITAPE	----	UNUSEDC00CC1	V	DUMMY	JDA	----	UNUSEDC0000E	V	DUMMY
JFM	----	UNUSEDC00C0F	V	DUMMY	JMB	----	UNUSEDC00CCD	V	DUMMY	JYR	----	UNUSEDC00003	V	DUMMY
NH	----	UNUSEDC00C05	V	DUMMY	NX	----	UNUSEDC00CC2	V	DUMMY	NY	----	UNUSEDC0000B	V	DUMMY
NZ	----	UNUSEDC00C04	V	DUMMY	RLAT	----	UNUSEDC00CCA	V	DUMMY	RLONG	----	UNUSEDC0000B	V	DUMMY

LOCAL VARIABLES (1 WORD):

C00C GETST

BLANK COMMON (0 WORDS)

ENTRY POINTS:

C00C GETST

EXTERNAL SUBPROGRAMS REQUIRED:

9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	20	C0014
CONSTANTS:	0	C00C
LOCAL VARIABLES:	1	C00C1
TEMPS:	17	C0011
TOTAL PROGRAM:	38	C0026

```

1.      SUBROUTINE GETV(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATW,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHM,IE0D)
3.      C      VERSION OF 29 JUNE 1971
4.      C
5.      C      SUBROUTINE GETV,  READS WORLD VOLCANOE CATALOGUE COMPILATION
6.      C
7.          DIMENSION PLT(5)
8.          DIMENSION ID(5)
9.      400      IIN = 105
10.         II0LT = 108
11.         IE0D=C
12.         KGDA=0
13.      410      CONTINUE
14.         READ(ITAPE,74) IAREA,IC0M,ISUBA,IDASH,IN0,LAT,RLATM,KNS,
15.         1  LONG,RL0M,KEW,IHEIT,IPT,IPAGE,ITYPE,ICHEM,ID
16.      74      FORMAT(I3,A1,I2,A1,I2,1X,I2,F5.2,A1,1X,I3,F5.2,A1,1X,
17.         1  I5,1X,I2,1X,I3,1X,A1,1X,A1,15X,5A4)
18.         CALL STAT(I)
19.         CALL EVIL(II0LT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0)
20.         IF(IBAD)410,53,900
21.      53      CONTINUE
22.         KGM0=IAREA
23.         KGYR=ISUBA
24.         KGHM=IN0
25.         CALL NAVIN(LAT,RLATM,KNS,LONG,RL0M,KEW,RLAT,RLONG,
26.         KGDA0=KGDA
27.         KGM00=KGM0
28.         KGYR0=KGYR
29.         KGHM0=KGHM
30.         PLT(1) = IN0
31.         PLT(2) = IHEIT
32.         PLT(3) = (IPT*1000) + IPAGE
33.         IF(NX)80,85,80
34.      80      DATAX=PLT(NX)
35.      85      DATAY=PLT(NY)
36.         DATAZ=IHEIT
37.         DATAW=IN0
38.         RETURN
39.      900      IE0D=1
40.         RETURN
41.         END

```

LABEL	HEX L8C	HEX L8C	HEX L8C	HEX L8C	HEX L8C
---	---	---	---	---	---
53	CC057	CC057	85	CC07F	410
9CC	CC085	CC07C	80	CC03F	400
		---	---	---	---
		HEX L8C	HEX L8C	HEX L8C	HEX L8C
		---	---	---	---
		74	00032	0007C	0001B

LOCAL VARIABLES (35 WORDS):

CCCCC	GETV	CCOC1	PLT	0006C	ID	0C0CB	IIN	0000C	IBUT	0000D	IAREA
CCCCC	ICBM	CCCCF	ISUEA	0001C	IDASH	00011	INB	00012	LAT	00013	RFLATM
CCCCC	KNS	GC015	LBNQ	00016	RLBN	00017	KEM	00018	IHEIT	00019	IPT
CCCC1A	IPAGE	0001B	ITYFE	0001C	ICHEN	0001D	I	0001E	IBAD	0001F	KGDAB
CCC2C	KGM8B	CCC21	KGYRB	00022	KGM8B						

ELANK COMMON (C WORDS)

ENTRY POINTS:

COCOC GETV

EXTERNAL SUBPROGRAMS REQUIRED:

EVIL	STAT	F:101	F:103	F:105	SBCDREAD	910DATA
918LUSA	9SETURN					
SITER						
NAVIN						

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	140	C008C
CONSTANTS:	0	C0C0C
LOCAL VARIABLES:	35	C0C23
TEMPS:	17	C0C11
	-----	-----
TOTAL PROGRAM:	192	C0CCC

```

1.      SUBROUTINE GETX(ITAPE,NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAW,
2.      1  RLAT,RLONG,KGDA,KGM0,KGYR,KGHM,IE0D)
3.      C
4.      C      VERSION OF 26 SEPT 1972, DUMMY ROUTINE
5.      C  SUBROUTINE GETX, FOR READING VARIABLE DATA INPUT, WRITE YOUR
6.      C      OWN GETX ROUTINE
7.      DIMENSION PLT(10)
8.      IIN = 105
9.      II0LT=108
10.     IE0D=C
11.     KGDA=0
12.     KGM0=C
13.     OUTPLT ' SUBROUTINE GETX IS A DUMMY ROUTINE, CALL EXIT'
14.     CALL EXIT
15.     RETURN
16.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
DATA*	---	---	---	---	DATA*	---	---	---	---	DATA*	---	---	---	---
DATAZ	---	---	---	---	EXIT	---	---	---	---	GETX	---	---	---	---
GETX	---	---	---	---	IE9D	---	---	---	---	IIN	---	---	---	---
IIBLT	---	---	---	---	ITAFE	---	---	---	---	KGDA	---	---	---	---
KGHP	---	---	---	---	KGMS	---	---	---	---	KGYP	---	---	---	---
NW	---	---	---	---	NX	---	---	---	---	NY	---	---	---	---
NZ	---	---	---	---	PLT	---	---	---	---	RLAT	---	---	---	---
RLONG	---	---	---	---		---	---	---	---		---	---	---	---

## LOCAL VARIABLES (13 WORDS):

NAME	TYPE	CLASS	HEX L8C	DEC WORDS
COCCC GETX	---	---	00001 PLT	10

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

COCCC GETX

## EXTERNAL SUBPROGRAMS REQUIRED:

NAME	TYPE	CLASS	HEX L8C	DEC WORDS
EXIT	---	---	F:108	1
SEND18L	---	---	9PRINT	1
SSETUPN	---	---	0000C IIBUT	1

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

NAME	TYPE	CLASS	HEX L8C	DEC WORDS
GENERATED CODE:	---	---	49	1
CONSTANTS:	---	---	0	1
LOCAL VARIABLES:	---	---	13	1
TEMPS:	---	---	17	1
TOTAL PROGRAM:	---	---	79	1

```

1. C SUBROUTINE GETY OF MARCH 19, 1972
2. C MODIFIED MARCH 19, 1972 BY F8LINSBEE TO READ NEW CGSDATA EPICENTER
3. C FORMAT - ONLY READS IN THE NECESSARY VALUES
4. C SUBROUTINE GETY(ITAPE,NX,NY,NZ,NW,CATAX,CATAY,CATZ,DATAW,
5. C 1 RLAT,RLONG,KDA,KMB,KYR,KHM,IEOD)
6. C SN AND WE WERE MADE INTO INTEGERS FOR COMPATIBILITY WITH THE SIG-7
7. C USAGE OF ALPHA NUMERICS
8. C INTEGER SN,WE
9. C DIMENSION PLT(5)
10. C DATA IFLAG/0/
11. IF(IFLAG=1) 400,410,40C
12. 400 IIN = 105
13. IIBUT = 108
14. IFLAG=1
15. KL=0
16. 41C CONTINUE
17. IEOD=C
18. 411 CONTINUE
19. READ(ITAPE,65)
20. * KDA,KMB,KYR,KHM ,DLAT,KSX,DLON,KWE, DEPT, AMAG
21. CALL STAT(I)
22. CALL EVIL (IIBUT,I,IIBAD,KGDA8,KGM88,KGYR8,KGHM8)
23. IF( IIBAD) 411,53,9CC
24. 53 CONTINUE
25. 65 FORMAT (6X,I2,I2,I2,I4,3X, F5.3,A1,F6.3,A1,F3.0,F3.2)
26. KGDA8=KDA
27. KGM88=KMB
28. KGYR8=KYR
29. KGHM8=KHM
30. CALL DNAV(DLAT,KSX,DLON,KWE,RLAT,RLONG,KL)
31. PLT(1)=KDA*10000+KMB*100+KYR
32. PLT(1)=(MTH*1000)+(KDA*10)+IY
33. PLT(2)=DEPT
34. PLT(3)=AMAG
35. IF(NX)80,85,8C
36. 8C CATAX=PLT(NX)
37. 85 CATAY=PLT(NY)
38. CATZ=DEPT
39. CATW=AMAG
40. RETURN
41. 9CC IEOD=1
42. RETLRN
43. END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
AMAG	R	SCALR	0000F V	1	DATAX	R	SCALR	*0002C V	DUMMY	DEPT	R	SCALR	*0001D V	DUMMY
DATAY	R	SCALR	*0001E V	1	DATZ	R	SCALR	*0001F V	DUMMY	DNAV	R	SPRGG	0000E V	1
CLAT	R	SCALR	*0001A V	1	DLBN	R	SCALR	0000C P	1	GETY	R	SPRGG	00000 V	1
EVIL	I	SPRGG	EXTERNAL		IBAD	I	SCALR	00011 V	1	IE9D	I	SCALR	*00027 V	DUMMY
I	I	SCALR	00010 V	1	IBUT	I	SCALR	00017 V	1	I	I	SCALR	00008 V	1
IFLAG	I	SCALR	*00018 V	1	IN	I	SCALR	00015 V	1	I	I	SCALR	*00023 V	DUMMY
ITAPE	I	SCALR	00012 V	1	IV	I	SCALR	*00026 V	DUMMY	KDA	I	SCALR	00013 V	1
KGDAB	I	SCALR	00014 V	1	KMP	I	SCALR	00016 V	1	KL	I	SCALR	00009 V	1
KGYR	I	SCALR	*00024 V	DUMMY	KSN	I	SCALR	00017 V	1	KME	I	SCALR	0000D V	1
KP8	I	SCALR	*00025 V	DUMMY	MTM	I	SCALR	00018 V	1	NW	I	UNUSED	*0001C V	DUMMY
KYR	I	SCALR	*00019 V	DUMMY	NY	I	SCALR	*00021 V	DUMMY	NZ	I	UNUSED	*00018 V	DUMMY
AX	I	SCALR	00001 V	5	RLAT	R	SPRGG	EXTERNAL		RLNG	R	SCALR	*00022 V	DUMMY
PLT	R	ARRAY	00001 V	5	STAT	I	SPRGG	EXTERNAL		WE	I	UNUSED		
SN	I	UNUSED												

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
53	0003D	80	00073	400	00016
411	0002C			410	0001E

## LOCAL VARIABLES (24 WORDS):

0000C GETY	00001 PLT
0000A CLAT	0000B KSN
0001C I	00011 IBAD
00016 MTM	00017 IV

BLANK COMPGN (C WORDS)

ENTRY POINTS:

0000C GETY

EXTERNAL SUBPROGRAMS REQUIRED:

DNAV	EVIL	STAT	F:101	F:103	F:105	98C0READ	910DATA
91TOR	SSETUPN						

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE	CONSTANTS	LOCAL VARIABLES	TEMP	TOTAL PROGRAM
129	24	18	171	171

```

1.      SUBROUTINE GINST(ITAPE,JTAPE,KK,KGCA,KGM6,
2.      1  KGYR,KGMM,IDIF,ISRC,RLAT,RLONG,ELEV,K977,0BSG,
3.      2  IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
4.      VERSION 8 MAY 1975, ADD INITIALIZATION ZEROS AND
5.      OUTPUT COMMENTS ON GRAVITY FORMULA
6.      VERSION OF 11 DEC 1974, TO ADD HANDLING OF PROJ4 OUTPUT
7.      VERSION 1 OCT 1974, FOR 1967 GRAVITY FORMULA
8.      VERSION OF 26 JULY 1973, START CONVERSION TO 1967 G FORMULA
9.      VERSION OF 26 APR 72
10.     TEMPORARY MODIF DUE TO BACKWARD CODE / MANAGE
11.     VERSION OF 27 DECEMBER 71
12.     VERSION OF 22 DECEMBER 1971
13.     GINST VERSION NOV 12,1971
14.     INPUT , NEW GSUM FORMAT *
15.     PREVIOUS GSUM FORMAT
16.     + 3 KEYS : LTKEY , LGKEY , IAKY
17.     OUTPUT , NEW GSUM FORMAT
18.     MODIF ON NOV 12,1971 BY MONGET J.M. TO INCLUDE :
19.     - USE OF DATA LOCATION TABLE
20.     - USE OF LABEL TAPES
21.     MOD AUG 16 1971 BY FOLINSBEE TO CORRECT ERROR ON CALLING ARGS OF GBLKI
22.     VERSION OF JUNE 29,71 DOES NOT WRITE EOF ONTO OUTPUT DEVICE
23.     MODIFIED JUNE 28 TO READ( OR WRITE) BLOCKED DATA BY A FOLINSBEE
24.     VERSION OF APRIL 16 TO OPTIONALLY SUPPRESS REWIND OF ITAPE AND JTAPE
25.
26.
27.     SUBROUTINE GINST, FOR GSUM FORMATTED DATA
28.
29.
30.     VERSION WITH DESIGNATION OF INPUT AND OUTPUT MAGNETIC TAPES
31.     BY USE OF SUBROUTINE MOUNT
32.
33.
34.     SSW(12) UP TO LIST DATE IDENTIFICATION
35.     SSW(26) UP TO OUTPUT ON HIGH SPEED PRINTER ONLY
36.     SSW(27) UP TO SUPPRESS REWIND OF TAPES AT START OF JOB
37.     * 1 SUPPRESS REWIND OF ITAPE
38.     * 2 SUPPRESS REWIND OF JTAPE
39.     * 9 SUPPRESS RWIND OF BOTH ITAPE AND JTAPE
40.     SSW(29) = 1 - TO READ AND TEST FOR SELECTED SOURCE CODE
41.     NUMBERS TO BE PROCESSED
42.     * 2 - TO READ AND TEST FOR SELECTED SOURCE CODES
43.     NUMBERS TO BE SKIPPED
44.     SSW(30) UP FOR INPUT DATA ON CARDS
45.     SSW(31) UP TO OUTPUT DATA ON CARDS
46.     SSW(40) UP TO PROCESS WITH BOUNDS USING CLT
47.     * 0 - PROCESS WITHOUT BOUNDS
48.     * 1 - PROCESS WITH BOUNDS USING THE DATA LOCATION TABLE
49.     SSW(60) UP TO PROCESS ONLY DATA WITH IFFC=4, ABTRACTOR OUTPUT
50.     SSW(61) UP TO REPLACE FA,BG,ELEV,LAT, LONG WITH AVERAGED VALUES
51.
52.     USES ENDIO(DUMMY), EVIL, STAT, ISW
53.     ASSUME ISW AND STAT INITIALIZED IN MAIN PROGRAM
54.
55.     DIMENSION IDTIN(20),IBK(20),ITK(20),IDENS(20),IDESC(17,20)
56.     DIMENSION IDTOT(20),JBK(20),JTK(20),JDENS(20),JDESC(17,20)
57.     DIMENSION PLT(20),IA(35),IAFMT(9),IASW(35),ISRC(16)
58.     DATA ITERI,ITER0,'EITP','E0TP' /
59.

```

GIN80000

GIN80040

GIN80050

GIN80080

GIN80090

GIN80100

GIN80110

GIN80120

GIN80130

GIN80140

GIN80150

GIN80180

GIN80240

GIN80250

GIN80260

GIN80270

GIN80280

GIN80290

GIN80300

GIN80310

GIN80320

GIN80330

GIN80360

```

60.      IF(KK)420,400,410
61.      C
62.      C -----      GSUM INITIALISATION LOGIC
63.      C
64.      400      IIN = 105
65.              IIOBT = 108
66.              IFLN=106
67.              IDISC = 100
68.      C
69.      DEGRA=1.745329E-2
70.      RADEG=57.29578
71.      C      NEF = N8. 8F FILE NOW BEING PROCESSED
72.              NEF = 1
73.              IFILE = 1
74.      C      NRECT = N8. 8F RECORDS NOW WRITTEN ON PRESENT OUTPUT TAPE
75.              NRECT = NZERO
76.      C
77.      OUTPLT 'GINBT 8F 8 MAY 1975'
78.      MAXCT = 125000
79.      ILI=101 ; IDL=100
80.      IF(ISW(40).NE.C)CALL ENCLT(J,CLATO,CLAB8,CLBLE,CLORI,IDL,ILI,0)
81.      * ; CALL SETSKP(INDICA) ; IDLT=0
82.      * ; NEF=1 ; IFILE=J-1
83.      OUTPLT NEF,IFILE
84.      NZERO=0
85.      KGDA8=NZERO
86.      KGM88=NZERO
87.      KGYR8=NZERO
88.      KGM8=NZERO
89.      IREC1 = 1
90.      IREC2=2
91.      IRECIN = 0
92.      C
93.      C -----
94.      C -----      CHECK SSW(29) TO SEE IF SOURCE CODE NUMBERS
95.      C -----      ARE TO BE READ FOR DATA SELECTION
96.      C
97.      407 IF(ISW(29).EG.C)GO TO 1405
98.      READ(IIN,900)ISRC
99.      900 FORMAT(16I5)
100.      IF(ISW(29).EG.1)WRITE(IIOBT,912)ISRC,GO TO 1405
101.      WRITE(IIOBT,913)ISRC
102.      913 FORMAT(1H0,10X,'SKIPPED SOURCE CODES = ',16I5)
103.      912 FORMAT(1H0,10X,'SELECTED SOURCE CODES = ',16I5)
104.      C
105.      C -----
106.      C
107.      1405 IF(ISW(40).NE.C)GO TO 720
108.      IF(ISW(30))404,404,720
109.      404      J=1
110.      405 READ (IIN,406) IDTIN(J),IBK(J),ITK(J),IDENS(J),
111.      1          (IDESC(K,J),K=1,17)
112.      406 FORMAT(A4,1X,A1,1X,11,1X,13,17A4)
113.      IF(IDTIN(J).NE.ITERI) J=J+1; GO TO 405
114.      NEF=1
115.      IFILE=J = 1
116.      OUTPLT NEF,IFILE
117.      720 IF(ISW(31))408,408,1410
118.      408      J=1
119.      409 READ (IIN,406) IDTOT(J),JBK(J),JTK(J),JDENS(J),

```

GIN80370

GIN80380

GIN80390

GIN80400

GIN80410

GIN80530

GIN80520

GIN80540

GIN80430

GIN80470

GIN80480

GIN80490

GIN80500

GIN80510

GIN80550

GIN80560

GIN80580

GIN80590

GIN80600

GIN80610

GIN80620

GIN80630

GIN80640

GIN80650

GIN80670

GIN80680

```

120.      1      (JDESC(K,J),K=1,17)
121.      IF (IDTST(J).NE.ITER8) J=J+1; GO TO 409
122.      NEND=1
123.      JFILE=J - 1
124.      OUTPUT NEND,JFILE
125.      1410 IF (ISW(30))1412,1412,1414
126.      1412 CONTINUE
127.      IF (ISW(40).EG.C)GOTO810
128.      READ (IDISC,406)IDTIN(1),IBK(1),ITK(1),IDENS(1),
129.      * (IDESC(K,1),K=1,17)
130.      810 IF (IDTIN(1).EG.ITER1)GOTO1414
131.      CALL MBUNT(ITAPE,IDTIN(1))
132.      WRITE (IIOUT,1413) IDTIN(1),IBK(1),ITK(1),IDENS(1),
133.      1      (IDESC(K,1),K=1,17)
134.      1413 FORMAT (1X,A4,1X,A1,1X,I1,1X,I3,17A4)
135.      IF (ISW(27).EG.1 .OR. ISW(27).EG.9) GOTO 2414
136.      REWIND ITAPE
137.      2414 CONTINUE
138.      1414 IF (ISW(31))1416,1416,1418
139.      1416 CONTINUE
140.      IF (IDTST(1).EG.ITER8) GO TO 1418
141.      CALL MBUNT(ITAPE,IDTST(1))
142.      WRITE (IIOUT,1413) IDTST(1),IBK(1),ITK(1),IDENS(1),
143.      1      (JDESC(K,1),K=1,17)
144.      IF (ISW(27).EG.2 .OR. ISW(27).EG.9) GO TO 1418
145.      REWIND ITAPE
146.      1418 CONTINUE
147.      RETURN
148.      C
149.      C ----- GSUM INPUT LOGIC
150.      C
151.      410 CONTINUE
152.      50 CALL ENDIG
153.      IF (ISW(30).EG.1) GO TO 100
154.      IF (ISW(40).NE.C) GO TO 700
155.      52 READ (ITAPE,11)IREC1,ISBRC,KGDA,KGMB,KGYR,KGHM,
156.      1      DLAT,DLONG,ELEV,K977,GBSG,IDEF,FA,BG,TC,IELC,IGC,
157.      2      RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
158.      GO TO 101
159.      100 CONTINUE
160.      READ(IIN,469)IREC1,ISBRC,KGDA,KGMB,KGYR,KGHM,
161.      1      DLAT,DLONG,ELEV,K977,GBSG,IDEF,FA,BG,TC,IELC,IGC,
162.      2      RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
163.      101 CONTINUE
164.      CALL STAT(I)
165.      710 CONTINUE
166.      CALL EVIL(IIOUT,I,IBAD,KGDA8,KGM88,KGYR8,KGHM8)
167.      IF (IBAD) 50, 53, 575
168.      C
169.      C      CHECKING IF IREC = 2 OR 1
170.      C
171.      53 IF (IREC1.NE.2) GO TO 600
172.      IF (IREC1.NE.C) OUTPUT 'INPLT ALREADY IN 1967 GRAV FORMULA'
173.      IRECIN = 1
174.      GO TO 70
175.      600 IF (IREC1=1)601,610,601
176.      601 IF (IREC1.EG.8)KK=8;RETURN
177.      IF (IREC1.EG.9)KK=9;RETURN
178.      GO TO 50
179.      C

```

GIN80690  
GIN80700  
GIN80710  
GIN80720  
GIN80730  
GIN80740  
GIN80750

GIN80770  
GIN80780  
GIN80790  
GIN80800  
GIN80810  
GIN80820  
GIN80830  
GIN80840  
GIN80850  
GIN80860  
GIN80870  
GIN80880  
GIN80890  
GIN80900  
GIN80910  
GIN80920  
GIN80930

GIN80940  
GIN80950  
GIN80960

GIN81020  
GIN81030

GIN81090  
GIN81100  
GIN81110  
GIN81120

```

180. C      CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF
181. C      THE 1967 INTERNATIONAL GRAVITY FORMULA
182. C      AND NEW GEODETIC REFERENCE SYSTEM
183. C
184. 610 CONTINUE
185. IF (IRECIN.EG.0) OUTPUT 'CONVERTING TO 1967 GRAV FORMULA NOW'
186. IRECIN = 1
187. KK = 1
188. CALL BG (K977,0BSG,GBBS,KK)
189. GBBS=GBBS+14.0
190. KK = 2
191. CALL BG (K977,0BSG,GBBS,KK)
192. RLAT = DLAT*DEGRA
193. DG=3.2*(13.6*(SIN(ABS(RLAT))*2))
194. IF(FA-990.0) 611,612,612
195. 611 FA=FA+DG
196. 612 IF(BG-990.0) 613,70,70
197. 613 BG=BG+DG
198. GO TO 70
199. 575 IF (NEF = IFILE) 576, 577, 577
200. 576 NEF = NEF + 1
201. IF (ISW(40).EG.0)GOTO820
202. READ(ICISC,406)IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
203. * (IDESC(K,NEF),K=1,17)
204. 820 CALL MOUNT(ITAPE,IDTIN(NEF))
205. WRITE (IIOUT,1413) IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
206. 1 (IDESC(K,NEF),K=1,17)
207. REWIND ITAPE
208. GO TO 50
209. C      END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
210. 577 KK=9 ; RETURN
211. 70 IF (ISW(12))73,73,71
212. 71 WRITE(IIOUT,72)KGDA,KGM0,KGYR,KGHM
213. 72 FORMAT('DATE=',3I3,15)
214. 73 IF (ISW(29).EG.0)GOTO1730
215. IF (ISW(29).EG.2)GOTO1700
216. C
217. C ----- PROCESS ONLY SELECTED SOURCE CODES
218. C
219. C01650=1,16
220. IF (ISRC(J).EG.0)GOTO50
221. IF (IS0RC-ISRC(J))1650,1730,1650
222. 1650 CONTINUE
223. GOTO50
224. C
225. C ----- IGNORE SELECTED SOURCE CODES
226. C
227. 1700 C01710=1,16
228. IF (ISRC(J).EG.0)GOTO1730
229. IF (IS0RC-ISRC(J))1710,50,1710
230. 1710 CONTINUE
231. C
232. C -----
233. C
234. 1730 RLAT=DLAT*DEGRA
235. RL0NG = DL0NG*DEGRA
236. IF (ISW(60))418,418,800
237. C USE DATA ONLY FOR IFFC = 4 (IE. ABSTRACTER OUTPUT)
238. 800 IF (IFFC=4)50,801,50
239. 801 CONTINUE

```

GIN01240

GIN01250

GIN01270

GIN01280

GIN01290

GIN01300

GIN01310

GIN01390

GIN01400

GIN01410

GIN01440

GIN01450

GIN01460

```

240.      IF (ISW(61)) 109, 109, 105
241. C   SET FA, BG, AND ELEV = AVERAGED VALUES FROM ABSTRACTER OUTPUT      GIN01480
242. C   SET LAT AND LONG TO VALUES AT CENTER OF GRID AREA                  GIN01490
243.      105 CONTINUE                                                         GIN01500
244.      DO 802 JK=1, 35                                                       GIN01510
245.      802 IASH(JK)=ISL(IA(JK), -24)                                       GIN01520
246.      CALL PKBY(IASH, IAFMT, 35)                                           GIN01530
247.      DEC0DE(35, 803, IAFMT) CLAT, CLONG, AHEIGT, KAFA, KABG             GIN01540
248.      803 FORMAT(2F9.6, F7.0, 2I5)                                         GIN01550
249.      RLAT=CLAT*DEGRA
250.      RLONG=CLONG*DEGRA
251.      AFA=FLOAT(KAFA)*0.1                                                    GIN01580
252.      ABG=FLOAT(KABG)*0.1                                                    GIN01590
253.      FA=AFA                                                                GIN01600
254.      BG=ABG                                                                GIN01610
255.      ELEV=AHEIGT                                                            GIN01620
256.      NUM0=IFBC                                                             GIN01630
257.      109 CONTINUE                                                         GIN01640
258.      418 CONTINUE                                                         GIN01650
259.      KGDA0=KGDA                                                            GIN01660
260.      KGM00=KGM0                                                            GIN01670
261.      KGYR0=KGYR                                                            GIN01680
262.      KGHM0=KGHM                                                            GIN01690
263.      RETURN                                                                GIN01700
264. C
265. C ----- GSUM OUTPUT LOGIC
266. C
267.      42C CALL ENDIO                                                         GIN01710
268.      CLAT=RLAT*RADEG
269.      CLONG=RLONG*RADEG
270.      PLAT=CLAT*90. / LTKEY=PLAT
271.      PLONG=CLONG*180. / LGKEY=PLONG
272.      CALL AREAK(DLAT, DLONG, IAKY)
273.      IF (ISW(31).EQ.1) GO TO 110                                           GIN01720
274.      IF (JTAPE.EQ.108) IREC2=C
275.      IF (ISW(26).EQ.1) IREC2=C, JTAPE=108
276.      WRITE(JTAPE, 11) IREC2, IS0RC, KGDA, KGM0, KGYR, KGHM,
277.      1 CLAT, CLONG, ELEV, K977, 0BSG, ICEP, FA, BG, TC, IELC, IGC,
278.      2 RFA, IREGC, IFFC, IA, IFBC, LTKEY, LGKEY, IAKY
279.      11 FORMAT(1I, 14, 3I2, 14, 2F9.4, F7.2, 13, F6.2, 15, 2F6.1, F4.1,
280.      1 2I2, F6.1, 1I, 12, 35A1, 1X, 1I, 2I3, 12)
281.      NRECT = NRECT + 1                                                       GIN01800
282.      IF (NRECT = MAXCT) 85, 90, 90                                         GIN01810
283.      85 RETURN                                                             GIN01820
284.      11C CONTINUE                                                           GIN01830
285.      WRITE(IPLN, 469) IREC2, IS0RC, KGDA, KGM0, KGYR, KGHM,
286.      1 CLAT, CLONG, ELEV, K977, 0BSG, ICEP, FA, BG, TC, IELC, IGC,
287.      2 RFA, IREGC, IFFC, IA, IFBC, LTKEY, LGKEY, IAKY
288.      469 FORMAT(1I, 14, 3I2, 14, 2F9.4, F7.2, 13, F6.2, 15, 2F6.1, F4.1,
289.      1 2I2, F6.1, 10X, 1I, 12, 35A1, 1X, 1I, 2I3, 12)
290.      RETURN
291.      9C NRECT = NZERO                                                         GIN01870
292.      WRITE (II0UT, 91)                                                       GIN01880
293.      91 FORMAT ('MAXCT OUTPUT')                                             GIN01890
294.      END FILE JTAPE                                                         GIN01900
295.      REWIND JTAPE                                                            GIN01910
296.      NEND=NEND+1                                                             GIN01920
297.      IF (NEND=JFILE) 990, 990, 995                                         GIN01930
298.      99C CALL MOUNT(JTAPE, IDT0T(NEND))                                     GIN01940
299.      WRITE (II0UT, 1413) IDT0T(NEND), WBK(NEND), WTK(NEND), JDENS(NEND),  GIN01950

```

```

300.      1      (JDESC(K,NEND),K=1,17)
301.      REWIND TAPE
302.      GO TO 999
303.      995  WRITE(IIOUT,902)NEND
304.      902  FORMAT('END EXCEEDS FILE,  NEND = ', I6)
305.      KK=9 ; RETURN
306.      999  CONTINUE
307.      500  RETURN
308.      C
309.      C -----  GSUM DLT INPUT LOGIC
310.      C
311.      700  CONTINUE
312.      IF(IDLT.EQ.1)GOTO821
313.      READ(ICISC,812)NMAX ; IPRE = NMAX ; NPRE=1
314.      812  FORMAT(I6)
315.      322  CONTINUE
316.      READ(ICISC,321,END=331)NBL0,ILAST,LASTR,0LMAX,0LMIN
317.      321  FORMAT(4X,I6,I6,I3,2F9.4)
318.      IF(NBL0.EQ.0)IDLT=0 ; GOTO575
319.      ALMIN=FLOAT(LASTR-90)
320.      ALMAX=ALMIN+1.
321.      0LARG=0LMAX-0LMIN
322.      DL0UP=DL0RI+0LARG
323.      DL0D0=DL0LE-0LARG
324.      IF((DL0UP.GE.0LMAX).AND.(DL0D0.LE.0LMIN))GOTO323
325.      IPRE=ILAST
326.      GOTO322
327.      323  CL0UP=DLAT0+1.
328.      CLAD0=CLAB0-1.
329.      IF((CL0UP.GE.ALMAX).AND.(CLAD0.LE.ALMIN))GOTO324
330.      IF(ALMAX.LT.CLAD0)GOTO343
331.      IPRE=ILAST
332.      GOTO322
333.      324  IF(IPRE.EQ.NMAX)GOTO325
334.      IBEG=IPRE+1
335.      GOTO326
336.      325  IBEG=1
337.      C -----  NINF = INDEX FIRST REC. TO READ
338.      C -----  NSUP = INDEX LAST REC. TO READ
339.      326  NFIR = NMAX*(NBL0-1)
340.      NINF=NFIR+IBEG
341.      NSUP=NFIR + ILAST
342.      C -----  AVOID TRYING TO READ REC. WHICH ARE ALREADY PROCESSED
343.      IF(NINF.LT.NPRE)NINF=NPRE
344.      NSKIP=NINF-NPRE
345.      CALL SKPREC(ITAPE,NSKIP,'FWD')
346.      GOTO(330,330,331,332,333)INDICA
347.      GOTO(330,331,332,333)INDICA
348.      330  CONTINUE
349.      NPRE=NSUP+1
350.      ICNREC = NSUP-NINF +1 ; ICNT = 0
351.      821  IF(ICNT.LT.ICNREC)IDLT=1 ; GOTO870
352.      IDLT=0 ; GOTO322
353.      870  READ(ITAPE,11)IREC1,IS0RC,KGDA,KGMB,KGYR,KGHM,
354.      1  CLAT,CLONG,ELEV,K977,0BSG,IDEF,FA,BG,TC,IELC,IGC,
355.      2  RFA,IREGC,IFFC,IA,IFBC,LTKEY,LGKEY,IAKEY
356.      ICNT = ICNT+1
357.      GOTO101
358.      343  WRITE(IIOUT,344)
359.      344  FORMAT(1H,'TEST AREA ALREADY PROCESSED',/)

```

GIN01980  
GIN01990  
GIN02000  
GIN02010  
GIN02020  
  
GIN02040  
GIN02050

```
360.      KK=9 ; RETURN
361. 331 WRITE(IIOUT,345)
362. 345 FORMAT(1H0,'INCORRECT DLT TABLE - FOUND EOF WHILE PROCESSING
363.      * RECORDS')
364.      KK=9 ; RETURN
365. 333 WRITE(IIOUT,346)
366. 346 FORMAT(1H0,'INCORRECT DLT TABLE - FOUND END OF TAPE WHILE
367.      * SKIPPING RECORDS',/)
368.      KK=9 ; RETURN
369. 332 WRITE(IIOUT,347)
370. 347 FORMAT(1H0,'ERROR CONDITION WHILE SKIPPING RECORDS',/)
371.      KK=9 ; RETURN
372.      END
```

GIN82060

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
ABG	R	SCALR	003C9 V	1	ABS	R	SPR0G	INTRIN	1	GINBT	R	SCALR	00000 V	DUMMY	AF	R	SCALR	003C8 V	1
AHEIGHT	R	SCALR	003C5 V	1	ALPAX	R	SCALR	003D6 V	1	IA	R	SCALR	00000 V	DUMMY	ALMIN	R	SCALR	003D5 V	1
AREAK	R	SPR0G	EXTERN	1	BG	R	SCALR	*003F3 V	DUMMY	IASH	I	ARRAY	*003FA V	DUMMY	CLAT	R	SCALR	003C3 V	1
CLOG	R	SCALR	003C4 V	1	DEGRA	R	SCALR	003F1 V	1	IBK	I	ARRAY	00015 V	20	DG	R	SCALR	003C1 V	1
DLAB0	R	SCALR	003AA V	1	DLA0B	R	SCALR	003D8 V	1	IDENS	I	ARRAY	00015 V	20	DLAT	R	SCALR	003B9 V	1
DLAT0	R	SCALR	003A9 V	1	DLAUP	R	SCALR	003D9 V	1	IDF	I	ARRAY	00030 V	DUMMY	DLDB	R	SCALR	003D9 V	1
DL0LE	R	SCALR	003AB V	1	DUBNG	R	SCALR	003DA V	1	IDT	I	SCALR	003AE V	DUMMY	DLORI	R	SCALR	003D9 V	1
DLOLP	R	SCALR	003AD V	1	ELEV	R	SCALR	*003EE V	DUMMY	IEFC	I	SCALR	003AF V	DUMMY	DLORI	R	SCALR	003D9 V	1
ENDLT	R	SPR0G	EXTERN	1	EVIL	R	SPR0G	EXTERN	1	IFILE	I	SCALR	003A2 V	1	ENDT0	R	SCALR	003AC V	1
FL0AT	R	SPR0G	INTRIN	1	GINBT	R	SCALR	00000 P	1	IBUT	I	SCALR	003A2 V	1	FA	R	SCALR	*003F2 V	DUMMY
GBBS	R	SCALR	003C0 V	1	IKEY	R	SCALR	003BE V	1	IELC	I	SCALR	003A2 V	1	GINBT	R	SCALR	00000 V	1
IAFMT	I	ARRAY	003BD V	1	IBEG	R	SCALR	003BD V	1	INDICA	I	SCALR	003A2 V	1	IA	R	SCALR	*003FA V	DUMMY
IBAD	I	SCALR	003BF V	1	IDONT	R	SCALR	003DC V	1	IREC1	I	SCALR	003A2 V	1	IASH	I	ARRAY	*003FA V	35
ICNREC	I	SCALR	003E1 V	1	IDESC	I	ARRAY	003E2 V	1	IREC2	I	SCALR	003A2 V	1	IBK	I	ARRAY	00015 V	20
IDISC	I	SCALR	*003F1 V	DUMMY	IDLT	I	SCALR	003A7 V	1	ISBRC	I	SCALR	003A2 V	1	IDENS	I	ARRAY	00015 V	20
IDTIN	I	SCALR	0039E V	1	IFFC	I	ARRAY	003A5 V	DUMMY	ITAPE	I	SCALR	003A2 V	1	IDF	I	ARRAY	00030 V	DUMMY
IFBC	I	SCALR	*003FB V	DUMMY	IN	I	SCALR	003B8 V	1	ITK	I	SCALR	003A2 V	1	IDT	I	SCALR	003AE V	1
IGC	I	SCALR	*003F6 V	DUMMY	IPUN	I	SCALR	003A6 V	1	ITR	I	SCALR	003A2 V	1	IELC	I	SCALR	003A2 V	1
ILAST	I	SCALR	003D1 V	1	IREC2	I	SCALR	003B4 V	1	ITR	I	SCALR	003A2 V	1	IBUT	I	SCALR	003A2 V	1
IPRE	I	SCALR	003D1 V	1	ISBRC	I	SCALR	003B4 V	1	ITR	I	SCALR	003A2 V	1	IREC1	I	SCALR	003A2 V	1
IREC1	I	SCALR	003B3 V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	IREC2	I	SCALR	003A2 V	1
ISL	I	SPR0G	INTRIN	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	IREC1	I	SCALR	003A2 V	1
ISW	I	SCALR	0039A V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	*003F8 V	DUMMY
ITER0	I	SCALR	0039A V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
JK	I	ARRAY	001E9 V	20	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
FILE	I	SCALR	003B8 V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
JTK	I	ARRAY	001CD V	20	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KAF	I	SCALR	003C6 V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KGM	I	SCALR	*003E9 V	DUMMY	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KGM00	I	SCALR	003B0 V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KK	I	SCALR	*003E5 V	DUMMY	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
LGKEY	I	SCALR	003BC V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
POINT	I	SCALR	EXTERN	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
NEND	I	SCALR	003B7 V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
NMAX	I	SCALR	003CD V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
NSKIP	I	SCALR	003E0 V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
NZER0	I	SCALR	003A4 V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
PLARG	R	SCALR	003D7 V	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
PKBY	R	SCALR	EXTERN	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
PLT	R	SCALR	00349 V	20	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
PLAT	R	SCALR	*003EC V	DUMMY	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
SIN	R	SPR0G	INTRIN	1	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
TC	R	SCALR	*003F4 V	DUMMY	ITR	I	SCALR	003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
ABG	R	SCALR	003C9 V	1	ABS	R	SPR0G	INTRIN	1	GINBT	R	SCALR	00000 V	DUMMY	AF	R	SCALR	003C8 V	1
AHEIGHT	R	SCALR	003C5 V	1	ALPAX	R	SCALR	003D6 V	1	IA	R	SCALR	00000 V	DUMMY	ALMIN	R	SCALR	003D5 V	1
AREAK	R	SPR0G	EXTERN	1	BG	R	SCALR	*003F3 V	DUMMY	IASH	I	ARRAY	*003FA V	DUMMY	CLAT	R	SCALR	003C3 V	1
CLOG	R	SCALR	003C4 V	1	DEGRA	R	SCALR	003F1 V	1	IBK	I	ARRAY	00015 V	20	DG	R	SCALR	003C1 V	1
DLAB0	R	SCALR	003AA V	1	DLA0B	R	SCALR	003D8 V	1	IDENS	I	ARRAY	00015 V	20	DLAT	R	SCALR	003D9 V	1
DLAT0	R	SCALR	003A9 V	1	DLAUP	R	SCALR	003D9 V	1	IDF	I	ARRAY	00030 V	DUMMY	DLDB	R	SCALR	003D9 V	1
DLOLE	R	SCALR	003AB V	1	DUBNG	R	SCALR	003DA V	1	IDT	I	SCALR	003AE V	DUMMY	DLORI	R	SCALR	003D9 V	1
DLOLP	R	SCALR	003AD V	1	ELEV	R	SCALR	*003EE V	DUMMY	IEFC	I	SCALR	003AF V	DUMMY	DLORI	R	SCALR	003D9 V	1
ENDLT	R	SPR0G	EXTERN	1	EVIL	R	SPR0G	EXTERN	1	IFILE	I	SCALR	003A2 V	1	ENDT0	R	SCALR	003AC V	1
FL0AT	R	SPR0G	INTRIN	1	GINBT	R	SCALR	00000 P	1	IBUT	I	SCALR	003A2 V	1	IA	R	SCALR	*003FA V	DUMMY
GBBS	R	SCALR	003C0 V	1	IKEY	R	SCALR	003BE V	1	IELC	I	SCALR	003A2 V	1	GINBT	R	SCALR	00000 V	1
IAFMT	I	ARRAY	003BD V	1	IBEG	R	SCALR	003BD V	1	IREC1	I	SCALR	003A2 V	1	IASH	I	ARRAY	*003FA V	35
IBAD	I	SCALR	003BF V	1	IDONT	R	SCALR	003DC V	1	IREC2	I	SCALR	003A2 V	1	IBK	I	ARRAY	00015 V	20
ICNREC	I	SCALR	003E1 V	1	IDESC	I	ARRAY	003E2 V	1	ISBRC	I	SCALR	003A2 V	1	IDENS	I	ARRAY	00015 V	20
IDISC	I	SCALR	*003F1 V	DUMMY	IDLT	I	SCALR	003A7 V	1	ISBRC	I	SCALR	003A2 V	1	IDF	I	ARRAY	00030 V	DUMMY
IDTIN	I	SCALR	0039E V	1	IFFC	I	ARRAY	003A5 V	DUMMY	ITAPE	I	SCALR	003A2 V	1	IBK	I	ARRAY	00015 V	20
IFBC	I	SCALR	*003FB V	DUMMY	IN	I	SCALR	003B8 V	1	ITK	I	SCALR	003A2 V	1	IDENS	I	ARRAY	00015 V	20
IGC	I	SCALR	*003F6 V	DUMMY	IPUN	I	SCALR	003A6 V	1	ITR	I	SCALR	003A2 V	1	IEFC	I	SCALR	003A2 V	1
ILAST	I	SCALR	003D1 V	1	IREC2	I	SCALR	003B4 V	1	ITR	I	SCALR	003A2 V	1	IBUT	I	SCALR	003A2 V	1
IPRE	I	SCALR	003D1 V	1	ISBRC	I	SCALR	003B4 V	1	ITR	I	SCALR	003A2 V	1	IREC1	I	SCALR	003A2 V	1
IREC1	I	SCALR	003B3 V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	*003F8 V	DUMMY
ISL	I	SPR0G	INTRIN	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
ISW	I	SCALR	0039A V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
ITER0	I	SCALR	0039A V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
JK	I	ARRAY	001E9 V	20	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
FILE	I	SCALR	003B8 V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
JTK	I	ARRAY	001CD V	20	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KAF	I	SCALR	003C6 V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KGM	I	SCALR	*003E9 V	DUMMY	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KGM00	I	SCALR	003B0 V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
KK	I	SCALR	*003E5 V	DUMMY	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
LGKEY	I	SCALR	003BC V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
POINT	I	SCALR	EXTERN	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
NEND	I	SCALR	003B7 V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
NMAX	I	SCALR	003CD V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16
NSKIP	I	SCALR	003E0 V	1	ITR	I	SCALR	*003E3 V	DUMMY	ITR	I	SCALR	003A2 V	1	ISRC	I	SCALR	00389 V	16

00000	GINBT	00001	IDTIN	00015	IBK	00029	ITK	0003D	IDENS	00051	IDESC
00001	JBK	00018	JTK	001CC	JTK	001E1	JDEIS	001F5	JDESC	00349	PLT
00015	IASH	00346	IASH	00389	ISRC	00399	ITERI	0039A	IN 39A	0039B	IN 39A
0003C	IBUT	0039C	IPUN	00399	IDIC	0039F	DEGRA	003A0	RADEG	003A1	NEF
0003A	I FILE	003A3	NRECT	003A4	NZERB	003A5	MAXCY	003A6	ILI	003A7	IDL 3A7
0003A	J	003A9	DLAT0	003AB	DLAT0	003AB	DL0LE	003AC	DL0RI	003AD	INDICA
0003A	IDLT	003AF	KGBA0	003BC	KGBH0	003B1	KGYRS	003B2	KGHM0	003B3	I REC1
0003A	I REC2	003B5	I RECIN	003BC	K	003B7	NEND	003B8	JFILE	003B9	DLAT
0003A	DLONG	003B8	LKEY	003BC	LKEY	003BC	NEND	003BE	I	003BF	IBAD
0003C	G8B	003C1	DG	003C2	JK	003C3	CLAT	003C4	CLONG	003C5	ANEIGT
0003C	KAF	003C7	KABG	003C8	AFA	003C9	ABG	003CA	NLM0	003CB	PLAT
0003C	PLONG	003CD	NMAX	003CE	IPRE	003CF	NPRE	003D0	NBL0	003D1	LAST
0003D	LASTR	003D3	BLMAX	003D4	BLMIN	003D5	ALMIN	003D6	ALMAX	003D7	BLARG
0003D	DLUP	003D8	DLUP	003CA	DLUP	003CB	DLAD0	003DC	IBEG	003DD	NPIR
0003D	NINP	003DF	NSUP	003EC	NSUP	003E1	ICNREC	003E2	IC0NT	003E3	IC0NT

### ENTRY POINTS:

COCCO GINOT

ABS	FL0AT	ISL	SIN
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	42
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	51	51	51
52	52	52	52
53	53	53	53
54	54	54	54
55	55	55	55
56	56	56	56
57	57	57	57
58	58	58	58
59	59	59	59
60	60	60	60
61	61	61	61
62	62	62	62
63	63	63	63
64	64	64	64
65	65	65	65
66	66	66	66
67	67	67	67
68	68	68	68
69	69	69	69
70	70	70	70
71	71	71	71
72	72	72	72
73	73	73	73
74	74	74	74
75	75	75	75
76	76	76	76
77	77	77	77
78	78	78	78
79	79	79	79
80	80	80	80
81	81	81	81
82	82	82	82
83	83	83	83
84	84	84	84
85	85	85	85
86	86	86	86
87	87	87	87
88	88	88	88
89	89	89	89
90	90	90	90
91	91	91	91
92	92	92	92
93	93	93	93
94	94	94	94
95	95	95	95
96	96	96	96
97	97	97	97
98	98	98	98
99	99	99	99
100	100	100	100

AREAK	END10	ENDLT	EVL	IS*	YOUNT	0BG	PXBY
SETSKP	SKPREC	STAT	F:101	F:102	F:103	F:104	F:105
F:106	9C8	9C8DEE	9C8DEAD	9C8WRIT	9C8ODE	SENDFILE	9END10L
910CAT	910R	910R	9PRINT	9REMINC	9RT91	9SETUPN	9

+HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	1331	DEC	WORDS
CONSTANTS:	24	HEX	WORDS
LOCAL VARIABLES:	95		
TEMP:	27		
TOTAL PROGRAM:	2377		

```

1.      FUNCTION GINTF(RLAT)
2.      C
3.      C      GINTF CALCULATES THEORETICAL GRAVITY FROM INTERNATIONAL
4.      C      GRAVITY FORMULA 'EARTH AND ITS GRAVITY FIELD' HEISKANEN
5.      C      AND VENING MEINESZ 1958 PAGE 74.      EXPANDED BY A.
6.      C      FOLINSBEE USING FORMULA  $(\sin(X))^2 = (1 - \cos(2X))/2$ 
7.      C
8.      C      VALUE RETURNS A G = 977000 MGALS
9.      C
10.     A = ABS(RLAT)
11.     GINTF = 3632.272-2586.157*cos(2.*A)+2.885*cos(4.*A)
12.     RETURN
13.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
GINTF	R	SCALR	CCCC00	V	1	ABS	GINTF	00000	P	C0S				
A	R	SCALR	CCCC00	V	1	GINTF				RLAT	R	SCALR	000002	V
														DUMMY

## LOCAL VARIABLES (2 WORDS):

00000 GINTF 00001 A

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

00000 GINTF

## INTRINSIC SUBPROGRAMS USED:

ABS C0S

## EXTERNAL SUBPROGRAMS REQUIRED:

9C0S 9SETUP1

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	DEC WORDS	HEX WORDS
CONSTANTS:	20	CCCC14
LOCAL VARIABLES:	5	CCCC05
TEMPS:	2	CCCC02
	3	CCCC03
TOTAL PROGRAM:	30	CCCC1E

```

1.      FUNCTION GI67F(RLAT)
2.      C      VERSION OF 25 APR 75 TO REDUCE FORMALLA BY USING
3.      C      EXPRESSION  $\sin(X)^2 = (1 - \cos(2X))/2$ 
4.      C      CALCULATES THEORETICAL GRAVITY FROM THE INTERNATIONAL FORMULA
5.      C      ACCORDING TO RESOLUTION NO. 2 OF THE XIV TH GENERAL ASSEMBLY OF
6.      C      THE I.U.G.G. 1967
7.      C      FOR DETAILS OF THE FORMULA SEE PAGE 74 OF
8.      C      GEODETIC REFERENCE SYSTEM 1967
9.      C
10.     C      VALLE RETURNED AS G-977000 MGALS
11.     C
12.     A=ABS(RLAT)
13.     GI67F = 3621.9455 - 2592.9639*COS(2.*A) + 2.8683*COS(4.*A)
14.     RETURN
15.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
A	R	SCALR	C0001 V	1	CBS	R	SPR0G	INTRIN		CBS	R	SPR0G	INTRIN	
G167F	R	SCALR	C0000 V	1	G167F	R	SPR0G	C0000 P		RLAT	R	SCALR	*00002 V	DUMMY

## LOCAL VARIABLES (2 WORDS):

C000C G167F C0001 A

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

C000C G167F

## INTRINSIC SUBPROGRAMS USED:

ABS C0S

## EXTERNAL SUBPROGRAMS REQUIRED:

9C0S 9SETUP1

## HIGHEST ERROR SEVERITY: C (NO ERRORS)

GENERATED CODE:	DEC WORDS	HEX WORDS
CONSTANTS:	20	C0014
LOCAL VARIABLES:	5	C0005
TEMP:	2	C0002
	3	C0003
TOTAL PROGRAM:	30	C001E

```

1.      SUBROUTINE GRIDG(ZZ,HGT,XFAC,YFAC,TOP,BOT,DLEFT,RIGHT)
2.      C SUBROUTINE GRIDG, FOR GRAF3
3.      C ANNOTATES X AND Y SCALES
4.      C
5.      C SSW(8) = LP TO SUPPRESS ANNOTATION OF X AND Y SCALES
6.      C
7.      C      USES SUBROUTINES FOR CALCOMP AND ISW
8.      C      ASSUME ISW INITIALIZED IN MAIN PROGRAM
9.      C
10.     C
11.     CS      IIBLT=2
12.     CS      IIBLT = 108
13.     CS      WRITE(IIBLT,10)
14.     CS 10   FORMAT('SET PEN TO ORIGIN OF PLOT IN BOTH X AND Y,')
15.     CS      PAUSE 10
16.     CS      CALL WHERE(XORG,YORG)
17.     CS      CALL WHERE (XORG, YORG, RFAC)
18.     CS      CALL PLOT(XORG,YORG,-3)
19.     CS      CALL SYMB(0.0,0.0,0.14,3,0.0,-1)
20.     CS      CALL SYMB(0.0,0.0,0.14,3,0.0,-1)
21.     CS      IF(ISW(8))300,20,300
22.     C PLOTTING X AND Y SCALES EVERY INCH
23.     20     ANGC=0.0
24.     CS      JDEC=-1
25.     CS      ANGDI=0.0
26.     CS      KDEC=-1
27.     CS      TOPY=TOP/YFAC
28.     CS      BOTY=BOT/YFAC
29.     CS      DLEFX=DLEFT/XFAC
30.     CS      RIGTX=RIGHT/XFAC
31.     CS      AK=0.2*ZZ
32.     CS      A3=0.3*ZZ
33.     CS      XX=DLEFX
34.     CS      YY=BOTY
35.     CS      ANT=DLEFT
36.     CS      CALL PLOT(XX,YY,3)
37.     C PLOTTING TIC
38.     50     YT=YY+A3
39.     CS      CALL PLOT(XX,YT,2)
40.     CS      XT=XX-AK
41.     CS      YT=YY-AK
42.     CS      CALL NLMB(XT,YT,HGT,ANT,ANGC,JDEC)
43.     CS      CALL NLMB (XT, YT, HGT, ANT, ANGC, JDEC)
44.     CS      CALL PLOT(XX,YY,3)
45.     CS      IF(XX-RIGTX)100,200,200
46.     100    XX=XX+(1.0*ZZ)
47.     CS      CALL PLOT(XX,YY,2)
48.     CS      ANT=ANT+(XFAC*ZZ)
49.     CS      GO TO 50
50.     C PLOTTING BORDER OF GRID LIMITS
51.     200    XX=RIGTX
52.     CS      YY=BOTY
53.     CS      CALL PLOT(XX,YY,3)
54.     CS      YY=TOPY
55.     CS      CALL PLOT(XX,YY,2)
56.     CS      XX=DLEFX
57.     CS      CALL PLOT(XX,YY,2)
58.     CS      ANT=TOP
59.     CS      AX=(0.5*ZZ)

```

```

60.      AY=-(C.03*ZZ)
61.  C   PLOTTING TIC
62.      22C  XT=XX+A3
63.      CALL PLOT(XT,YY,2)
64.      XT=XX+AX
65.      YT=YY+AY
66.  CS   CALL NLMB(XT,YT,HGT,ANT,ANGD,KDEC)
67.      CALL NLMB(XT,YT,HGT,ANT,ANGD,KDEC)
68.      CALL PLOT(XX,YY,3)
69.      IF(YY-B0TY)300,300,250
70.      25C  YY=YY-(1.C*ZZ)
71.      CALL PLOT(XX,YY,2)
72.      ANT=ANT-(YFAC*ZZ)
73.      GO TO 220
74.      30C  RETURN
75.      END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
AK	R	SCALR	0000D V	1	ANGC	R	SCALR	00005 V	1	ANGD	R	SCALR	00007 V	1	AY	R	SCALR	00015 V	1
ANT	R	SCALR	00011 V	1	AX	R	SCALR	00014 V	1	BTY	R	SCALR	0000A V	1	GRIGD	R	SCALR	00000 V	1
A3	R	SCALR	0000E V	1	DLEFX	R	SCALR	0000B V	1	IIBUT	I	SCALR	00001 V	1	KDEC	I	SCALR	00008 V	1
CLEFT	R	SCALR	0001C V	DUMMY	HGT	R	SCALR	00017 V	DUMMY	REFACT	R	SCALR	00004 V	1	SYMBOL	R	SCALR	00000 V	1
GRIGD	I	SPRGG	00000 F		JDEC	I	SPRGG	00006 V	1	WHERE	R	SPRGG	00003 V	1	XT	R	SCALR	00013 V	1
ISW	I	SPRGG	00000 F		PLBT	R	SCALR	00009 V	1	YORG	R	SCALR	00003 V	1	ZZ	R	SCALR	00016 V	DUMMY
NUMBER	I	SPRGG	00000 F		RIGTX	R	SCALR	00002 V	DUMMY										
RIGHT	R	SCALR	0001D V	DUMMY	TOPY	R	SCALR	00009 V	1										
TOP	R	SCALR	0001A V	DUMMY	XORG	R	SCALR	00002 V	DUMMY										
XFAC	R	SCALR	00018 V	DUMMY	YFAC	R	SCALR	00019 V	DUMMY										
XX	R	SCALR	0000F V	1	YY	R	SCALR	00010 V	1										
YT	R	SCALR	00012 V	1															

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
20	00024	100	00067	220	00096
300	000C4	200	00075	250	000B4

## LOCAL VARIABLES (22 WORDS):

00000	GRIGD	00001	IIBUT
00006	JDEC	00007	ANGD
0000C	RIGTX	0000D	AK
00012	YT	00013	XT

## BLANK COMMON (C WORDS):

## ENTRY POINTS:

00000 GRIGD

## EXTERNAL SUBPROGRAMS REQUIRED:

ISW	NUMBER	PLBT	SYMBOL	WHERE	SSETUPN

## HIGHEST ERROR SEVERITY: C (NO ERRORS)

DEC WORDS	HEX WORDS
197	000C5
12	0000C
22	00016
9	00009
240	000F0

TOTAL PROGRAM: 240

```

1.      SUBROUTINE GRID2(ZZ,ZHT,NUMPL,DEGRA,FDEG2,RDEG2,RTOP,ITOP,RBOT,
2.      1 IBOT,RLEFT,ILEFT,RRIGT,IRIGT,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,
3.      2 NDEG,SLAT,SLONG,BOTMP)
4.      C
5.      C PLOTS AND ANNOTATES MERCATOR CHART GRID
6.      C 20 OCTOBER 1972
7.      C
8.      C
9.      C SSW(1) LP TO DRAW PERIMETER OF GRID ONLY
10.     C SSW(5) =0 TO MAKE DEGREE ANNOTATIONS INSIDE GRID (CHARACTER SIZE 0.07 INCH)
11.     C          =1 TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.21 INCH)
12.     C          =2 TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.35 INCH)
13.     C SSW(8) LP TO SUPPRESS PLOTTING OF GRID
14.     C
15.     C      USES SUBROUTINES WHR, ISW, AND CALCOMP ROUTINES
16.     C      ASSUMES ISW INITIALIZED IN MAIN PROGRAM
17.     C
18.     C      INITIALIZING DISTANCE AND CHARACTER HEIGHT CONSTANTS
19.     10  SA=0.02*ZZ
20.         SB=0.05*ZZ
21.         SC=0.15*ZZ
22.         SD=0.22*ZZ
23.         SE=0.18*ZZ
24.         SF=-0.25*ZZ
25.         SG=0.26*ZZ
26.         IF(ISW(5).EQ.0) ZFAC=1.0 GO TO 18
27.         IF(ISW(5).EQ.1) ZFAC=3.0
28.         IF(ISW(5).EQ.2) ZFAC=5.0
29.         TA=((0.05+(ZFAC*0.05))*ZZ
30.         TB=((0.07+(ZFAC*0.07))*ZZ
31.         TC=(ZFAC*0.24)*ZZ
32.         TD=0.04*ZZ
33.     18  HGT=ZFAC*0.07*ZHT
34.     C SET ORIGIN FOR CHART
35.         CALL WHERE(XX,YY,RFACT)
36.         CALL PLOT(XX,YY,-3)
37.         IF(ISW(8))80,20,80
38.     C ANNOTATING PLOT NUMBER IN LOWER LEFT HAND CORNER
39.     20  IF(ISW(5))121,21,121
40.     121 SF=((ZFAC+1.0)*0.24)*ZZ
41.     21  CALL SYMBOL(SF,SA,HGT,NUMPL,90.,4)
42.     C ANNOTATING GRID
43.         IF(ISW(5))24,24,22
44.     22  CALL NUMBER(TA,TB,HGT,FLEFT,0.0,-1)
45.         GO TO 26
46.     24  CALL NUMBER(SB,SA,HGT,FLEFT,0.0,-1)
47.     26  CALL PLOT(C.0,0.0,3)
48.     C START PLOTTING GRID
49.         SLAT=RBOT
50.         SLONG=RRIGT
51.         CALL WHR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMP,SLAT,SLONG,BOTMP,XX,YY)
52.         CALL PLOT(XX,YY,2)
53.         IF(ISW(5))30,30,28
54.     28  XT=XX+TA
55.         YT=YY+TB
56.         GO TO 32
57.     30  XT=XX+SG
58.         YT=YY+SB
59.     32  CALL NUMBER(XT,YT,HGT,FRIGT,0.0,-1)

```

```

60.      IF(ISW(5)) 71,71,72
61.      72 XT=XX-TA
62.      YT=TD
63.      GO TO 73
64.      71 YT=SD
65.      73 CALL NLMBR(XT,YT,HGT,FBOT,0.0,-1)
66.  C. DRAW LEFT AND TOP SIDES OF FIDUCIAL HALF-INCH SQUARE
67.      XFID=XX+1.0
68.      YFID=YY+0.5
69.      CALL PLBT(XFID,YFID,3)
70.      YFID=YFID+0.5
71.      CALL PLBT(XFID,YFID,2)
72.      XFID=XFID+0.5
73.      CALL PLBT(XFID,YFID,2)
74.      CALL PLBT(XX,YY,3)
75.      SLAT=RTOP
76.      CALL WRR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMP,SLAT,SLONG,BBTMP,XX,YY)
77.      CALL PLBT(XX,YY,2)
78.      IF(ISW(5)) 75,75,76
79.      76 YT=YY-TD
80.      GO TO 77
81.      75 YT=YY-SE
82.      77 CALL NLMBR(XT,YT,HGT,FBOT,0.0,-1)
83.      CALL PLBT(XX,YY,3)
84.      SLONG=RLEFT
85.      CALL WRR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMP,SLAT,SLONG,BBTMP,XX,YY)
86.      CALL PLBT(XX,YY,2)
87.      IF(ISW(5)) 36,36,34
88.      34 XT=TC
89.      YT=YY-TD
90.      GO TO 38
91.      36 XT=XX+SC
92.      38 CALL NLMBR(XT,YT,HGT,FBOT,0.0,-1)
93.      CALL PLBT(XX,YY,3)
94.      CALL PLBT(0.0,0.0,2)
95.      IF(ISW(5)) 42,42,40
96.      40 XT=TC
97.      YT=TD
98.      GO TO 44
99.      42 XT=SC
100.     YT=SD
101.     44 CALL NLMBR(XT,YT,HGT,FBOT,0.0,-1)
102.  C FINISHED DRAWING AND ANNOTATING PERIMETER OF GRID
103.  C CHECKING IF NDEG GRID LINES WANTED
104.     IF(ISW(1)) 80,50,80
105.     50 NND=((IRIGT-ILEFT)/NDEG)*NDEG
106.     NBOT=1
107.     DO 60 N=NDEG,NND,NDEG
108.     JK=IRIGT-(ILEFT+N)
109.     IF(JK) 51,61,51
110.     51 AAA=N
111.     RMORE=DEGRA*AAA
112.     SLONG=RLEFT+RMORE
113.     FLONG=SLONG*57.29578
114.     IF(NBOT=1) 54,52,54
115.     52 SLAT=RBOT
116.     CALL WRR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMP,SLAT,SLONG,BBTMP,XX,YY)
117.     CALL PLBT(XX,YY,3)
118.     IF(ISW(5)) 55,55,53
119.     53 XT=XX+TA

```

```

120.      YT=YY+TB
121.      GO TO 56
122.      55  XT=XX+SB
123.      YT=YY+SA
124.      56  CALL NUMBER (XT, YT, HGT, FLONG, 0.0, -1)
125.      CALL PLOT (XX, YY, 3)
126.      SLAT=RTOP
127.      CALL WHR (DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BOTMP, XX, YY)
128.      CALL PLOT (XX, YY, 2)
129.      NBOT=2
130.      GO TO 60
131.      54  SLAT=RTOP
132.      CALL WHR (DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BOTMP, XX, YY)
133.      CALL PLOT (XX, YY, 3)
134.      SLAT=RBOT
135.      CALL WHR (DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BOTMP, XX, YY)
136.      CALL PLOT (XX, YY, 2)
137.      IF (ISW(5)) 58, 58, 57
138.      57  XT=XX+TA
139.      YT=YY+TB
140.      GO TO 59
141.      58  XT=XX+SB
142.      YT=YY+SA
143.      59  CALL NUMBER (XT, YT, HGT, FLONG, 0.0, -1)
144.      CALL PLOT (XX, YY, 3)
145.      NBOT=1
146.      C FINISHED DRAWING AND ANNOTATING THIS LONGITUDE LINE
147.      60  CONTINUE
148.      C COMPLETED ALL LONGITUDE GRID LINES
149.      61  NND=((ITOP-IBOT)/NDEG)*NDEG
150.      NLEFT=2
151.      DO 70 N=NDEG, NND, NDEG
152.      JK= ITOP-(IBOT+N)
153.      IF (JK) 161, 80, 161
154.      161  AAA=N
155.      RMORE=DEGRA*AAA
156.      SLAT=RBOT+RMORE
157.      FLAT=SLAT*57.29578
158.      IF (NLEFT=1) 64, 62, 64
159.      62  SLONG=RLEFT
160.      CALL WHR (DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BOTMP, XX, YY)
161.      CALL PLOT (XX, YY, 3)
162.      IF (ISW(5)) 65, 65, 63
163.      63  XT=X+TC
164.      YT=YY+TD
165.      GO TO 66
166.      65  XT=XX+SC
167.      YT=YY+SB
168.      66  CALL NUMBER (XT, YT, HGT, FLAT, 0.0, -1)
169.      CALL PLOT (XX, YY, 3)
170.      SLONG=RRIGHT
171.      CALL WHR (DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BOTMP, XX, YY)
172.      CALL PLOT (XX, YY, 2)
173.      NLEFT=2
174.      GO TO 70
175.      64  SLONG=RRIGHT
176.      CALL WHR (DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BOTMP, XX, YY)
177.      CALL PLOT (XX, YY, 3)
178.      SLONG=RLEFT
179.      CALL WHR (DEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BOTMP, XX, YY)

```

```

180.      CALL PLOT(XX,YY,2)
181.      IF(ISW(5))68,68,67
182.      67  XT=XX-TC
183.          YT=YY-TD
184.          GO TO 69
185.      68  XT=XX+SC
186.          YT=YY+SB
187.      69  CALL NUMBER (XT, YT, HGT, FLAT, 0.0, -1)
188.          CALL PLOT(XX,YY,3)
189.          NLEFT=NLEFT-1
190.      C  FINISHED ANNOTATING AND DRAWING THIS LATITUDE LINE
191.      7C  CONTINUE
192.      C  COMPLETED ALL GRID LINES AND ANNOTATIONS
193.      8C  RETURN
194.          END

```

[illegible]

LOCAL VARIABLES (31 WORDS):!

0000C	GRID2	000C1 SA	000C2 SB	000C3 SC	00004 SD	00005 SE
0000C	SE	000C7 SG	000C8 ZFAC	000C9 TA	0000A TB	0000B TC
0000C	TD	000C0	000CE XX	000CF YY	00010 RFACT	00011 XT
00012 YT		000C3 XFID	00014 YFIC	000C5 NND	00016 N89T	00017 N
00018 XK		00015 AAA	0001A FMBRE	0001B FLBNG	0001C NLEFT	0001D FLAT
0001E X						

ELANK COMEN (C WORDS)

ENTRY POINTS:

00000 GRID2

EXTERNAL SUBPROGRAMS REQUIRED:

ISK	NUMBER	PLOT	SYMBOL	WHERE	WHR	GTR
-----	--------	------	--------	-------	-----	-----

HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	755	002F3
CONSTANTS:	26	0001A
LOCAL VARIABLES:	31	0001F
TEMPS:	27	0001B
TOTAL PROGRAM:	839	00347

```

1.      SUBROUTINE INCEP(SL,SW,CX,CY,CXP,CYP,BX,BY,A1,A2,B1,B2)
2.      C
3.      C  SUBROUTINE INCEP, DETERMINES INTERCEPT POINTS OF LINE BETWEEN
4.      C  ANY 2 POINTS AND GIVEN BOUNDARIES
5.      C
6.      C  USES SUBROUTINE EXTD
7.      C
8.      N=0
9.      SM=(CY-CYP)/(CX-CXP)
10.     B=((CYP-CX)-(CY-CXP))/(CX-CXP)
11. 20C IF((ABS(CXP))-SL)205,205,300
12. 205 IF((ABS(CYP))-SW)206,206,300
13. 206 A1=CXP
14.     B1=CYP
15.     N=N+1
16. 21C IF((ABS(CX))-SL)215,215,500
17. 215 IF((ABS(CY))-SW)220,220,500
18. 22C A2=CX
19.     B2=CY
20.     N=N+1
21.     IF(N=2)800,700,800
22. 30C BX=SL
23.     BY=SM*BX+B
24. 305 IF((ABS(BY))-SW)31C,31C,330
25. 31C CALL EXTD(CX,CXP,CY,CYP,BX,BY,IND)
26.     IF(IND)33C,320,330
27. 32C A1=BX
28.     B1=BY
29.     N=N+1
30. 33C BY=SW
31.     BX=(BY-B)/SM
32. 335 IF((ABS(BX))-SL)34C,34C,365
33. 34C CALL EXTD(CX,CXP,CY,CYP,BX,BY,IND)
34.     IF(IND)365,35C,365
35. 35C N=N+1
36.     IF(N=2)36C,355,800
37. 355 A2=BX
38.     B2=BY
39.     GO TO 700
40. 36C A1=BX
41.     B1=BY
42. 365 BX=-SL
43.     BY=SM*BX+B
44. 37C IF((ABS(BY))-SW)375,375,390
45. 375 CALL EXTD(CX,CXP,CY,CYP,BX,BY,IND)
46.     IF(IND)39C,385,390
47. 385 N=N+1
48.     IF(N=2)386,387,800
49. 386 A1=BX
50.     B1=BY
51.     GO TO 390
52. 387 A2=-SL
53.     B2=-SL
54.     B2=BY
55.     GO TO 700
56. 39C BY=-SW
57.     BX=(BY-B)/SM
58. 395 IF((ABS(BX))-SL)405,405,400
59. 405 CALL EXTD(CX,CXP,CY,CYP,BX,BY,IND)

```

```

60.      IF(IND)400,410,400
61.      400  IF(N-1)900,420,800
62.      420  A2=CX
63.      B2=CY
64.      GO TO 700
65.      410  N=N+1
66.      IF(N-2)411,412,800
67.      411  A1=BX
68.      B1=BY
69.      A2=CX
70.      B2=CY
71.      GO TO 700
72.      412  A2=BX
73.      B2=BY
74.      IF(N-2)800,700,800
75.      500  BX=SL
76.      BY=SM*BX+B
77.      505  IF((ABS(BY))-SW)515,515,525
78.      515  CALL EXTD(CX,CXP,CY,CYP,BX,BY,IND)
79.      IF(IND)525,520,525
80.      520  N=N+1
81.      A2=BX
82.      B2=BY
83.      IF(N-2)800,700,800
84.      525  BY=SW
85.      BX=(BY-B)/SM
86.      530  IF((ABS(BX))-SL)545,545,555
87.      545  CALL EXTD(CX,CXP,CY,CYP,BX,BY,IND)
88.      IF(IND)555,550,555
89.      550  N=N+1
90.      A2=BX
91.      B2=BY
92.      IF(N-2)800,700,800
93.      555  BX=-SL
94.      BY=SM*BX+B
95.      560  IF((ABS(BY))-SW)580,580,590
96.      580  CALL EXTD(CX,CXP,CY,CYP,BX,BY,IND)
97.      IF(IND)590,585,590
98.      585  N=N+1
99.      A2=BX
100.     B2=BY
101.     IF(N-2)800,700,800
102.     590  BY=-SW
103.     BX=(BY-B)/SM
104.     595  IF((ABS(BX))-SL)600,600,800
105.     600  N=N+1
106.     A2=BX
107.     B2=BY
108.     IF(N-2)800,700,800
109.     700  RETURN
110.     800  WRITE (108,802)
111.     802  FORMAT ('INCEP:  BAD BRANCH')
112.     900  A1 = 9999.0
113.     A2 = 9999.0
114.     B1 = 9999.0
115.     B2 = 9999.0
116.     RETURN
117.     END

```

NAME	TYPE	CLASS	HEX	DEC	NAME	TYPE	CLASS	HEX	DEC
			L0C	WORDS				L0C	WORDS
ABS	R	SPR0G	INTRIN		A1	R	SCALR	*000CD	V DUMMY
B	R	SCALR	Q0003	V DUMMY	BX	R	SCALR	*000CB	V DUMMY
B1	R	SCALR	*000F	V DUMMY	B2	R	SCALR	*00C10	V DUMMY
CB	R	SCALR	*0009	V DUMMY	CY	R	SCALR	*000CB	V DUMMY
EXP	R	SPR0G	EXTERN		INCEP	I	SCALR	G00CC	V 1
EXTD	I	SCALR	Q00C4	V 1	N	I	SCALR	G00CC1	V 1
IND	R	SCALR	Q00C2	V 1	SW	R	SCALR	*G00CC6	V DUMMY
IN									
					A2	R	SCALR	*0000E	V DUMMY
					BY	R	SCALR	*0000C	V DUMMY
					CX	R	SCALR	*00007	V DUMMY
					CYP	R	SCALR	*0000A	V DUMMY
					INCEP	R	SPR0G	00000	P
					SL	R	SCALR	*00005	V DUMMY

[illegible]

## LOCAL VARIABLES (5 WORDS):

0000C	INCEP	00001 N	000C2 SH	00003 B	00004 IND
-------	-------	---------	----------	---------	-----------

BLANK COMMON (C WORDS)

ENTRY POINTS:

COCOC INCEP

INTRINSIC SUBPROGRAMS USED:

**ABS**

EXTERNAL SUBPROGRAMS REQUIRED:

```
EXTD      F:108      9ENC10L      9PRINT      9SETUPN
```

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC	HEX
WORDS	WORDS
---	---
GENERATED CODE:	00152
CONSTANTS:	00001
LOCAL VARIABLES:	00005
TEMP:	0000F
---	---
TOTAL PROGRAM:	00167

```

1. FUNCTION ISW(I)
2. C VERSION OF 27 JULY 1973, OUTPUT SSW CHANGE IF MADE
3. C VERSION OF 25 JULY 1973, TO ADD ENTRY ICHG
4. C THIS FUNCTION READS A CARD WHEN I<0, SETTING VALUES OF ISW(I) IN COLUMNS 1 TO
5. C 79. THE VALUE OF ISW(0) IS SET EQUAL TO ISW(80). WITH THIS EXCEPTION
6. C THE VALUE OF ISW(I) CORRESPONDS TO THE COLUMN NUMBER.
7. C FOR I<0 RETURNS THE VALUE OF JSW(I)
8. DIMENSION JSW(0:80)
9. IIN=105
10. IIOU=108
11. IF (I.LT.0) GO TO 50
12. ISW=JSW(I)
13. RETURN
14. C
15. C ENTRY POINT TO ALLOW CHANGING OF SSW VALUE
16. C
17. ENTRY ICHG(I,IVAL)
18. JSW(I)=IVAL
19. WRITE(IIOU,42)I,IVAL
20. 42 FORMAT(' SENSE SWITCH ',I4,' SET = TO ',I2)
21. RETURN
22. 50 READ(IIN,60)(JSW(K), K=1,80)
23. 60 FORMAT(80I1)
24. JSW(0)=JSW(80)
25. OUTPUT ' ISW FUNCTION, VERSION OF 27 JULY 73'
26. WRITE(IIOU,70)(JSW(K), K=0, 79)
27. 70 FORMAT('SENSE SWITCH OPTIONS (ISW): ',8(10I1,1X),4X,'(0-79)')
28. RETURN
29. END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
I	I	SCALR	00055	V DUMMY	ICHG	I	SPR0G	000CF	P	IIN	I	SCALR	00052	V
IOUT	I	SCALR	00053	V	ISW	I	SPR0G	00000	P	ISW	I	SCALR	00000	V
I VAL	I	SCALR	00055	V DUMMY	JSH	I	ARRAY	00001	V	K	I	SCALR	00054	V

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
LABEL					LABEL				
42	0001C		50	00029	70	00036		00057	

## LOCAL VARIABLES (85 WORDS):

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
00000	ISW		00001	JSH	00052	IIN		00053	IISUT
									00054
									K

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

NAME	TYPE	CLASS	HEX LOC	DEC WORDS
00000	ISW		000CF	ICHG

## EXTERNAL SUBPROGRAMS REQUIRED:

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
F:101	9BCDWRIT		F:102	9END10L	F:103	910DATA		F:104	9SPRINT
								F:105	9SETUP1
								F:106	9SETUP2
								F:108	9BCDREAD

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

NAME	TYPE	CLASS	HEX LOC	DEC WORDS
GENERATED CODE:			104	00068
CONSTANTS:			0	00000
LOCAL VARIABLES:			85	00055
TEMPS:			3	00003
TOTAL PROGRAM:			192	00000

```

1.      SUBROUTINE M2DY(IY,M0,IDAY,ID)
2.      C***      M2DY CHANGES MON, DAY TO THE NUMBER OF DAYS FOR THAT YEAR
3.      C
4.      C***      IY=YEAR (CONVERTS FOR LEAP YEAR
5.      C***      M0=MONTH
6.      C***      IDAY=DAY
7.      C***      ID=OUTPUT WHICH IS THE PROGRESSIVE DAY NUMBER
8.      C
9.      DIMENSION MYDAY(12)
10.     DATA MYDAY/1,32,60,91,121,152,182,213,244,274,305,335/
11.     C***      DETERMINE IF LEAP YEAR
12.     A=IY
13.     B=IY/4
14.     A=A/4.0
15.     IF (A-B) 12,10,12
16.     10      LEAP=1
17.     GO TO 13
18.     12      LEAP= 0
19.     13      CONTINUE
20.     IF(M0-2) 20,21,22
21.     21      IF(IDAY-28) 20,20,22
22.     20      LEAP=0
23.     22      ID=IDAY+MYDAY(M0)-1+LEAP
24.     RETURN
25.     END

```

HEX	L0C	LABEL	HEX	L0C	LABEL	HEX	L0C	LABEL	HEX	L0C	LABEL	HEX	L0C	LABEL
00014	10	-----	00017	12	-----	00019	13	-----	00020	20	-----	0001D	21	-----
00015	11	-----	00018	13	-----	0001A	14	-----	00021	21	-----	0001E	22	-----
00016	12	-----	00019	14	-----	0001B	15	-----	00022	22	-----	0001F	23	-----
00017	13	-----	0001A	15	-----	0001C	16	-----	00023	23	-----	00020	24	-----
00018	14	-----	0001B	16	-----	0001D	17	-----	00024	24	-----	00021	25	-----
00019	15	-----	0001C	17	-----	0001E	18	-----	00025	25	-----	00022	26	-----
0001A	16	-----	0001D	18	-----	0001F	19	-----	00026	26	-----	00023	27	-----
0001B	17	-----	0001E	19	-----	00020	20	-----	00027	27	-----	00024	28	-----
0001C	18	-----	0001F	20	-----	00021	21	-----	00028	28	-----	00025	29	-----
0001D	19	-----	00020	21	-----	00022	22	-----	00029	29	-----	00026	30	-----
0001E	20	-----	00021	22	-----	00023	23	-----	0002A	30	-----	00027	31	-----
0001F	21	-----	00022	23	-----	00024	24	-----	0002B	31	-----	00028	32	-----
00020	22	-----	00023	24	-----	00025	25	-----	0002C	32	-----	00029	33	-----
00021	23	-----	00024	25	-----	00026	26	-----	0002D	33	-----	0002A	34	-----
00022	24	-----	00025	26	-----	00027	27	-----	0002E	34	-----	0002B	35	-----
00023	25	-----	00026	27	-----	00028	28	-----	0002F	35	-----	0002C	36	-----
00024	26	-----	00027	28	-----	00029	29	-----	00030	36	-----	0002D	37	-----
00025	27	-----	00028	29	-----	0002A	30	-----	00031	37	-----	0002E	38	-----
00026	28	-----	00029	30	-----	0002B	31	-----	00032	38	-----	0002F	39	-----
00027	29	-----	0002A	31	-----	0002C	32	-----	00033	39	-----	00030	40	-----
00028	30	-----	0002B	32	-----	0002D	33	-----	00034	40	-----	00031	41	-----
00029	31	-----	0002C	33	-----	0002E	34	-----	00035	41	-----	00032	42	-----
0002A	32	-----	0002D	34	-----	0002F	35	-----	00036	42	-----	00033	43	-----
0002B	33	-----	0002E	35	-----	00030	36	-----	00037	43	-----	00034	44	-----
0002C	34	-----	0002F	36	-----	00031	37	-----	00038	44	-----	00035	45	-----
0002D	35	-----	00030	37	-----	00032	38	-----	00039	45	-----	00036	46	-----
0002E	36	-----	00031	38	-----	00033	39	-----	0003A	46	-----	00037	47	-----
0002F	37	-----	00032	39	-----	00034	40	-----	0003B	47	-----	00038	48	-----
00030	38	-----	00033	40	-----	00035	41	-----	0003C	48	-----	00039	49	

LOCAL VARIABLES (16 WORDS):

000000 M2DY	000001 MYDAY	0000D A	000CE B	0000F LEAP
-------------	--------------	---------	---------	------------

BLANK COMMON (0 WORDS)

## ENTRY POINTS:

COCC R2DY

EXTERNAL SUBPROGRAMS REQUIRED:

9170R 9SETLFA

HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	---	---
CONSTANTS:	41	C0C29
LOCAL VARIABLES:	2	C0C02
TEMP:	16	C0C1C
	5	C0C05
	---	---
TOTAL PROGRAM:	64	C0C4C

```

1.      SUBROUTINE NAVIN(LAT,RLATM,KNS,LANG,RL0M,KEW,RLAT,RLONG)
2.      C
3.      C  SUBROUTINE NAVIN,  CONVERTS ANNOTATED DEGREES AND MINUTES
4.      C                      TO SIGNED RADIAN LATITUDE AND
5.      C                      LONGITUDE
6.      C
7.      C      USES SUBROUTINE DMT0R
8.      C
9.      C
10.     C
11.     NNS = 1HS
12.     NEW = 1HW
13.     CS  NNS=1238
14.     CS  NEW=1278
15.     6C  RLAT=DMT0R(LAT,RLATM)
16.     RL0NG=DMT0R(L0NG,RL0M)
17.     IF(KNS=NNS)75,7C,75
18.     C  SOUTH LATITUDE
19.     7C  RLAT=-RLAT
20.     75  IF(KEW=NEW)85,8C,85
21.     C  WEST LONGITUDE
22.     8C  RL0NG=-RL0NG
23.     85  CONTINUE
24.     RETURN
25.     END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
DMT0R	R	SPR0G	-----	-----	KEH	I	SCALR	*00008	V DUMMY	KNS	I	SCALR	*00005	V DUMMY
LAT	I	SCALR	*00003	V DUMMY	LONG	I	SCALR	*00006	V DUMMY	NAVIN	I	SCALR	00000	V
NAVIN	R	SPR0G	00000	F	NEW	I	SCALR	00002	V	NNS	I	SCALR	00001	V
RLAT	R	SCALR	*00009	V DUMMY	RLATM	R	SCALR	*00004	V DUMMY	RL0M	R	SCALR	*00007	V DUMMY
RL0NG	R	SCALR	*0000A	V DUMMY										

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
60 0000F	70 0001C	75 0001F	80 00022	85 00025	

## LOCAL VARIABLES (3 WORDS):

00000 NAVIN 00001 NNS 00002 NEW

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

00000 NAVIN

## EXTERNAL SUBPROGRAMS REQUIRED:

DMT0R 9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX LBC
GENERATED CODE:	38	00026
CONSTANTS:	2	00002
LOCAL VARIABLES:	3	00003
TEMPS:	9	00009
TOTAL PROGRAM:	52	00034

```

1.      SUBROUTINE NAVBT(RLAT,RLONG,LAT,RLATM,KNS,
2.      1LONG,RLBM,KEW,KDEC)
3.      C
4.      C SUBROUTINE NAVBT CONVERTS RADIAN TO DEG,MIN AND LETTER FOR HEMISPHERE
5.      C
6.      C      USES SUBROUTINES RTDMO, RTDM1, RTDM2
7.      C
8.      C
9.      KDEC=KDEC+1
10.     CS      JN=116B
11.     CS      JS=123B
12.     CS      JE=105B
13.     CS      JW=127B
14.     JS      JN = 1FN
15.     JS      JS = 1FS
16.     JS      JE = 1FE
17.     JS      JW = 1FW
18.     ALAT=ABS(RLAT)
19.     41C GO TO(420,421,422),KDEC
20.     42C CALL RTDMO(ALAT,LAT,RLATM)
21.     GO TO 429
22.     421 CALL RTDM1(ALAT,LAT,RLATM)
23.     GO TO 429
24.     422 CALL RTDM2(ALAT,LAT,RLATM)
25.     429 IF(RLAT)430,432,432
26.     43C KNS=JS
27.     GO TO 435
28.     432 KNS=JN
29.     435 ALONG=ABS(RLONG)
30.     GO TO(436,437,438),KDEC
31.     436 CALL RTDMO(ALONG,LONG,RLBM)
32.     GO TO 439
33.     437 CALL RTDM1(ALONG,LONG,RLBM)
34.     GO TO 439
35.     438 CALL RTDM2(ALONG,LONG,RLBM)
36.     439 IF(RLONG)440,442,442
37.     44C KEW=JW
38.     GO TO 445
39.     442 KEW=JE
40.     445 RETURN
41.     END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
ABS	R	SPR0G	00003	1	AL0NG	R	SCALR	00006	1	AL0NG	R	SCALR	00006	1
JE	I	SCALR	00004	1	JS	I	SCALR	00002	1	JS	I	SCALR	00002	1
KN	I	SCALR	00005	1	KEW	I	SCALR	00000	1	KEW	I	SCALR	00000	1
NAV0T	R	SPR0G	00006	1	LBNG	R	SCALR	00000	1	LBNG	R	SCALR	00000	1
RLATM	R	SCALR	00007	1	RLAT	R	SCALR	00007	1	RLAT	R	SCALR	00007	1
RTDM0	R	SCALR	00008	1	RL0NG	R	SCALR	00008	1	RL0NG	R	SCALR	00008	1
					RTDM2	R	SPR0G	00008	1	RTDM2	R	SPR0G	00008	1

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
410	00017	420	0001F	429	00030
432	00035	435	00037	438	00040
440	00054	442	00057	439	00052

## LOCAL VARIABLES (7 WORDS):

CO000 NAV0T 00001 1N  
CO006 AL0NG

## BLANK COMMON (0 WORDS)

## ENTRY PRINTS:

CO000 NAV0T

## INTRINSIC SUBPROGRAMS USED:

ABS

## EXTERNAL SUBPROGRAMS REQUIRED:

RTDMC RTDM1 RTDM2 9SETUPN

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS	HEX WORDS
90	CO05A
4	CO004
7	CO007
10	CO00A
111	CO06F

TOTAL PROGRAM: 111

```

1.      SUBROUTINE BBG(K977,G0GR,G0BS,KK)
2.      C  SUBROUTINE BBG, CHANGE OBSERVED GRAVITY IN TWO
3.      C  WORDS TO GRAVITY LESS 977000.0 IF
4.      C  KK=1, OR VISA VERSA FOR KK=2
5.      C
6.      C
7.      IF(KK)200,100,100
8.      C  TWO WORDS TO ONE
9.      100  A = K977-977
10.      G0BS=(A * 1000.0)+G0GR
11.      RETURN
12.      C  ONE WORD TO TWO
13.      200  A=G0BS*0.001
14.      II=A
15.      B=II
16.      G0GR=(A+B)*1000.0
17.      K977=II+977
18.      RETURN
19.      END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
A	R	SCALR	00001	V 1	B	R	SCALR	00003	V 1	G0BS	R	SCALR	00006	V DUMMY
G0GR	R	SCALR	+00005	V DUMMY	II	I	SCALR	00002	V 1	KK	I	SCALR	+00007	V DUMMY
K977	I	SCALR	+00004	V DUMMY	0BG	R	SCALR	00000	V 1	0BG	SPR0G		00000	P

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
100	00009	200	00012		

## LOCAL VARIABLES (4 WORDS):

00000 0BG	00001 A	00002 II	00003 B
-----------	---------	----------	---------

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

00000 0BG

## EXTERNAL SUBPROGRAMS REQUIRED:

91T0R 9RT0I 9SETUPN

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	DEC WORDS	HEX WORDS
CONSTANTS:	35	00023
LOCAL VARIABLES:	2	00002
TEMP:	4	00004
	5	00005
TOTAL PROGRAM:	46	0002E

```

1.      SUBROUTINE BLINE(ZZ,ZHT,
2.      A  IDATA,IEBD,IIN,IIBUT,ITAPE,NUMPL,CATA,RLAT,RLONG,K0GHM,IAGAP,LCN
3.      BT,RADEG,DEGRA,KDEG2,IDEG2,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,LEFT,
4.      C  ILEFT,RRIGT,IRIGT,SLTK,SLGK,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,NDEG,
5.      D  SLAT,SLONG,BOTMP,XX,YY,INIT,XOLD,YOLD)
6.      C
7.      C      VERSION J1 JULY 73 TO INCLUDE FIDUCIAL SQUARE
8.      C  SUBROUTINE BLINE(ZZ,ZHT), PLOTS GRID FOR MERCATOR
9.      C      CHART HAVING NON-INTEGER BOUNDARIES
10.     C
11.     C
12.     C      SSW(8) UP TO SUPPRESS PLOTTING OF GRID
13.     C
14.     C
15.     C      CALLS SUBROUTINES RTDMC, WHR, ISW, AND CALCOMP ROUTINES
16.     C
17.     C
18.     C
19.     C      INITIALIZING DISTANCE AND CHARACTER HEIGHT CONSTANTS
20.     C
21.     10  HGT=0.07*ZHT
22.         SA=0.02*ZZ
23.         SB=0.05*ZZ
24.         SC=0.15*ZZ
25.         SD=0.22*ZZ
26.         SE=0.18*ZZ
27.         SF=-0.25*ZZ
28.         SG=0.26*ZZ
29.         CALL RTDMC(RTOP,LAT,RLATM)
30.         CALL RTDMC(RLEFT,LANG,RLBM)
31.         FTOP=LAT
32.         FTOPM=RLATM
33.         FLEFT=LANG
34.         FLFTM=RLBM
35.         CALL RTDMC(RBOT,LAT,RLATM)
36.         CALL RTDMC(RRIGT,LANG,RLBM)
37.         FBOT=LAT
38.         FBOTM=RLATM
39.         FRIGT=LANG
40.         FRGTM=RLBM
41.     C  SET ORIGIN FOR CHART
42.         CALL WHERE(XORG,YORG,RFACT)
43.     CS   CALL WHERE(XORG,YORG)
44.         CALL PLOT(XORG,YORG,-3)
45.         IF(ISW(8))80,20,80
46.     C  ANNOTATING CONSECUTIVE PLOT NUMBER
47.     20  PLNUM=NUMPL
48.         CALL NUMBER(SF,SA,HGT,PLNUM,90,-1)
49.     CS   CALL NUMB(SF,SA,HGT,PLNUM,90,0,-1)
50.         NUMPL=NUMPL+1
51.     C  ANNOTATING GRID
52.     CS   CALL NUMB(SB,SA,HGT,FLEFT,0,0,-1)
53.         CALL NUMBER(SB,SA,HGT,FLEFT,0,0,-1)
54.         XT=SB+SD
55.         YT=SA
56.     CS   CALL NUMB(XT,YT,HGT,FLFTM,0,0,-1)
57.         CALL NUMBER(XT,YT,HGT,FLFTM,0,0,-1)
58.         CALL PLOT(0,0,0,0,3)
59.     C  START PLOTTING GRID

```

```

60.      SLAT=RBOT
61.      SLONG=RRIGT
62.      CALL WFR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMF,SLAT,SLONG,BOTMP,XX,YY)
63.      CALL PLOT (XX,YY,2)
64.      C  DRAW LEFT AND TOP SIDES OF FIDUCIAL SQUARE
65.      XFID=XX+1.0
66.      YFID=YY-0.5
67.      CALL PLOT(XFID,YFID,3)
68.      YFID=YFID+0.5
69.      CALL PLOT(XFID,YFID,2)
70.      XFID=XFID+0.5
71.      CALL PLOT(XFID,YFID,2)
72.      CALL PLOT(XX,YY,3)
73.      XT=XX-(2.0*SG)-SB
74.      YT=YY+SB
75.      CS  CALL NUMB(XT,YT,HGT,FRIGT,0.0,-1)
76.      CALL NUMB(XT,YT,HGT,FRIGT,0.0,-1)
77.      XT=XX-SG
78.      YT=YY+SB
79.      CS  CALL NUMB(XT,YT,HGT,FRGTM,0.0,-1)
80.      CALL NUMB(XT,YT,HGT,FRGTM,0.0,-1)
81.      CALL PLOT(XX,YY,3)
82.      SLAT=RTOP
83.      SLONG=RRIGT
84.      CALL WFR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMF,SLAT,SLONG,BOTMP,XX,YY)
85.      CALL PLOT (XX,YY,2)
86.      SLAT=RTOP
87.      SLONG=RLEFT
88.      CALL WFR(DEGRA,FDEG2,RDEG2,RLEFT,SINCH,SMF,SLAT,SLONG,BOTMP,XX,YY)
89.      CALL PLOT(XX,YY,2)
90.      XT=XX+SC
91.      YT=YY-SE
92.      CS  CALL NUMB(XT,YT,HGT,FTOP,0.0,-1)
93.      CALL NUMB(XT,YT,HGT,FTOP,0.0,-1)
94.      XT=XT+SG+SB
95.      CS  CALL NUMB(XT,YT,HGT,FTOPM,0.0,-1)
96.      CALL NUMB(XT,YT,HGT,FTOPM,0.0,-1)
97.      CALL PLOT (XX,YY,3)
98.      CALL PLOT(C*0.0,0.0,2)
99.      XT=SC
100.     YT=SD
101.     CS  CALL NUMB(XT,YT,HGT,FBOT,0.0,-1)
102.     CALL NUMB(XT,YT,HGT,FBOT,0.0,-1)
103.     XT=XT+SG+SB
104.     CS  CALL NUMB(XT,YT,HGT,FBOTM,0.0,-1)
105.     CALL NUMB(XT,YT,HGT,FBOTM,0.0,-1)
106.     OUTPUT 'SUBROUTINE BLINE VERSION 11 JULY 73'
107.     C  FINISHED PLOT AND ANOT BASIC GRID
108.     8C  RETURN
109.     END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS		
B0TTP	R	SCALR	*00040	V DUMMY	DATA	R	UNUSED*00021	V DUMMY	1	DEGRA	R	SCALR	*00028	V DUMMY		
F00T	R	SCALR	*00039	V DUMMY	FB0T	R	SCALR	*0000F	V 1	FDEG2	R	SCALR	*0002B	V DUMMY		
FLEFT	R	SCALR	*0003B	V DUMMY	FLFTM	R	SCALR	*0000E	V 1	FRGTM	R	SCALR	*00010	V 1		
FRTGT	R	SCALR	*0003C	V DUMMY	FT0P	R	SCALR	*0003A	V DUMMY	FT0PM	R	SCALR	*0000D	V DUMMY		
HGT	R	SCALR	*00018	V DUMMY	JAGAP	R	UNUSED*00025	V DUMMY	1	IB0T	R	UNUSED*00030	V DUMMY	1		
ICATA	R	UNUSED*0001D	V DUMMY	1	IDEG2	R	UNUSED*0002A	V DUMMY	1	IE0D	R	UNUSED*0001C	V DUMMY	1		
IIN	R	UNUSED*00043	V DUMMY	1	IIBLT	R	UNUSED*0001E	V DUMMY	1	ILEFT	R	UNUSED*00032	V DUMMY	1		
INIT	R	UNUSED*0001F	V DUMMY	1	IRIGT	R	UNUSED*00034	V DUMMY	1	ISW	R	SPR0G	EXTERN	1		
ITAPE	R	UNUSED*00024	V DUMMY	1	IT0P	R	UNUSED*0003E	V DUMMY	1	KDEG2	R	UNUSED*00029	V DUMMY	1		
K0GPM	R	UNUSED*00020	V DUMMY	1	LAT	R	SCALR	*00009	V DUMMY	1	LCNT	R	UNUSED*00026	V DUMMY	1	
LONG	R	SCALR	*00020	V DUMMY	NDEG	R	UNUSED*0003D	V DUMMY	1	NUMBER	R	SCALR	*00000	V DUMMY	1	
NUMFL	R	SCALR	*00014	V DUMMY	0LINE	R	SPR0G	*0000C	F	0LINE	R	UNUSED*00027	V DUMMY	1		
PLNLM	R	SCALR	*0002F	V DUMMY	PL0T	R	SPR0G	EXTERN	1	RADEG	R	SCALR	*00013	V DUMMY	1	
RB0T	R	SCALR	*00022	V DUMMY	RDEG2	R	SCALR	*0002C	V DUMMY	1	RFACT	R	SCALR	*00031	V DUMMY	1
RLAT	R	UNUSED*00022	V DUMMY	1	RLATM	R	UNUSED*00033	V DUMMY	1	RLEFT	R	SCALR	*00033	V DUMMY	1	
RL0M	R	SCALR	*0000C	V DUMMY	RL0NG	R	SCALR	*0002D	V DUMMY	1	RRIGHT	R	SCALR	*00002	V DUMMY	1
RTDPO	R	SPR0G	EXTERN	1	RT0P	R	SCALR	*00004	V DUMMY	1	SA	R	SCALR	*00005	V DUMMY	1
SB	R	SCALR	*00003	V DUMMY	SC	R	SCALR	*00007	V DUMMY	1	SD	R	SCALR	*00008	V DUMMY	1
SE	R	SCALR	*00037	V DUMMY	SF	R	SCALR	*0003E	V DUMMY	1	SG	R	UNUSED*00036	V DUMMY	1	
SINGH	R	SCALR	*0003F	V DUMMY	SLAT	R	UNUSED*00035	V DUMMY	1	SLGK	R	SCALR	*00038	V DUMMY	1	
SLONG	R	SPR0G	EXTERN	1	SLTK	R	SPR0G	EXTERN	1	SMP	R	SCALR	*00017	V DUMMY	1	
WHERE	R	UNUSED*00044	V DUMMY	1	WHR	R	SCALR	*00018	V DUMMY	1	XFID	R	SCALR	*00015	V DUMMY	1
X0LD	R	SCALR	*00041	V DUMMY	X0RG	R	SCALR	*00016	V DUMMY	1	XT	R	UNUSED*00045	V DUMMY	1	
XX	R	SCALR	*00012	V DUMMY	YFID	R	SCALR	*00019	V DUMMY	1	Y0LD	R	SCALR	*00042	V DUMMY	1
Y0RG	R	SCALR	*0001A	V DUMMY	YT	R	SCALR	*00017	V DUMMY	1	YY	R	SCALR	*00042	V DUMMY	1
ZHT	R	SCALR	*0001A	V DUMMY	ZZ	R	SCALR	*00019	V DUMMY	1						

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
10	0003C	80	00176		

## LOCAL VARIABLES (25 WORDS):

0000C 0LINE	00001 HGT
00006 SE	00007 SF
0000C RL6M	0000D FT0PM
00012 Y0RG	00013 RFACT
00018 YFID	

BLANK COPY8N (C WORDS)

ENTRY PRINTS:

COCCC 0LINE

EXTERNAL SUBPROGRAMS REQUIRED:

ISW SIT0R	NUMBER SFRINT	PL0T SSETUPN	RTDVC	WHERE	WHR	F:108	9ENDI0L

00004 SC	00005 SD
0000A RLATM	0000B LONG
00010 FRGTM	00011 X0RG
00016 YT	00017 XFID

HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	375	C0177
CONSTANTS:	19	C0013
LOCAL VARIABLES:	25	C0019
TEMPS:	46	C002E
	-----	-----
TOTAL PROGRAM:	465	C01D1

```

1.      SUBROUTINE PINOT(ITAPE,JTAPE,KK,ISTA,KEY,LAT,LATM,KNS,
2.      1 LONG,L0M,KEW,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,
3.      2 DINE,STHIK,CRVN,WGTN,AVWTN,CRVW,WGTW,AVWTW)
4.      C      VERSION OF 20 OCT 1975 TO ELIMINATE STAT,WHAT DIDN'T WORK ANYWAY
5.      C      VERSION OF 11 DEC 1974, TO ADD HANDLING OF PROJ4 OUTPUT
6.      C      VERSION OF 29 MAY 1973, CHECKS FOR CARDS OUT OF ORDER
7.      C      VERSION OF 9 JUNE 1972, REMOVES SETTING OF ITAPE AND JTAPE
8.      C      VERSION OF 14 MARCH 1972, ADDING IMANT TO ARG LIST
9.      C      VERSION OF 7 MARCH 1972, ADDING SORT KEYS
10.     C      VERSION OF 11 MARCH 1971
11.     C SUBROUTINE PINOT, FOR INPLT AND OUTPUT OF SEISMIC REFRACTION
12.     C      PROFILE DATA AT SPFMT FORMAT
13.     C
14.     C
15.     C SSW(26) LP TO SET JTAPE = 108 AND IREC1 = 2
16.     C SSW(32) LP TO READ SPFMT DATA ON TWO CARDS
17.     C SSW(33) LP TO WRITE SPFMT DATA ON TWO CARDS
18.     C
19.     C      DIMENSION          IDESC(6),VEL(8),THICK(8)
20.     C
21.     C      USES SUBROUTINE ISW,NAVIN
22.     C
23.     C *****
24.     C      IF(KK)420,400,410
25.     C
26.     C      ITAPE = URN FOR SEISMIC DATA INPLT
27.     C      JTAPE = URN FOR DATA OUTPUT
28.     C *****
29.     C
30.     400 CONTINUE
31.     IIN = 105
32.     IIOUT = 108
33.     ISTA=0
34.     IKEY=0
35.     RADEG=57.29578
36.     WRITE (IIOUT,600)
37.     600 FORMAT ( / 'SUBROUTINE PINOT, VERSION 20 OCT 1975' //)
38.     IF (ISW(26).EQ.1) WRITE(IIOUT,601);
39.     $OUTPUT 'IREC1 SET TO 2 TO AVOID SLEWING' ;
40.     $WRITE (IIOUT,601)
41.     601 FORMAT(' *****')
42.     RETURN
43.     C
44.     C READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
45.     410 CONTINUE
46.     10 CONTINUE
47.     IF (ISW(32))15,12,15
48.     12 READ (ITAPE,990,END=900,ERR=10)
49.     $      IREC1,ISTA,KEY,LAT,LATM,KNS,L0NG,L0M,
50.     1 KEW,U1,K1,U2,K2,U3,K3,U4,K4,U5,K5,U6,K6,U7,K7,U8,K8,
51.     2 IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,STHIK,CRVN,
52.     3 WGTN,AVWTN,CRVW,WGTW,AVWTW
53.     990 FORMAT(I1,I4,A1,I2,I2,A1,I3,I2,A1,8(I2,I3),I2,I4,4I1,
54.     1 I1,I2,6A2,1X,2F4.1,F3.1,2F6.0,1X,F3.1,2F6.0,5X,2I3,I2)
55.     GO TO 18
56.     15 READ (ITAPE,991,END=900,ERR=10)
57.     $      IREC1,ISTA,KEY,LAT,LATM,KNS,L0NG,L0M,
58.     1 KEW,U1,K1,U2,K2,U3,K3,U4,K4,U5,K5,U6,K6,U7,K7,U8,K8,
59.     2 IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,STHIK,CRVN,

```

```

60.      3  WGTN,AVWTN,CRVW,WGTW,AVWTW
61.    991  FORMAT(I1,I4,A1,I2,I2,A1,I3,I2,A1,8(I2,I3),I2,I4,4I1,
62.      1  I1,I2/10X,6A2,1X,2F4.1,F3.1,2F6.0,1X,F3.1,2F6.0,5X,2I3,I2)
63.    18  CONTINUE
64.      IF(ISW(32))22,30,22
65.  C      CHECK FOR CARDS OUT OF ORDER
66.    22  IF(IREC1=1)24,30,24
67.    24  IF(IREC1.EQ.8)KK=8;RETURN
68.      OUTPUT 'IREC1 .NE. 1'
69.      OUTPUT ISTA8
70.      READ(ITAPE,25)
71.    25  FORMAT(1X)
72.      GO TO 15
73.    30  VEL(1)=(FL0AT(J1))*0.1
74.      VEL(2)=(FL0AT(J2))*0.1
75.      VEL(3)=(FL0AT(J3))*0.1
76.      VEL(4)=(FL0AT(J4))*0.1
77.      VEL(5)=(FL0AT(J5))*0.1
78.      VEL(6)=(FL0AT(J6))*0.1
79.      VEL(7)=(FL0AT(J7))*0.1
80.      VEL(8)=(FL0AT(J8))*0.1
81.      THICK(1)=(FL0AT(K1))*0.1
82.      THICK(2)=(FL0AT(K2))*0.1
83.      THICK(3)=(FL0AT(K3))*0.1
84.      THICK(4)=(FL0AT(K4))*0.1
85.      THICK(5)=(FL0AT(K5))*0.1
86.      THICK(6)=(FL0AT(K6))*0.1
87.      THICK(7)=(FL0AT(K7))*0.1
88.      THICK(8)=(FL0AT(K8))*0.1
89.      ISTA8=ISTA
90.    98  RETURN
91.  C
92.  C  OUTPUT RECORD
93.  C
94.    42C  CONTINUE
95.      LTKEY=LAT+90
96.      RLATM=LATM
97.      RL0M=L0M
98.      CALL NAVIN(LAT,RLATM,KNS,L0NG,RL0M,KEW,RLAT,RL0NG)
99.  C  CALCULATING SORTING KEYS
100.      PLAT=(RLAT*RADEG)+90
101.      PL0NG=(RL0NG*RADEG)+180
102.      LTKEY=PLAT
103.      LGKEY=PL0NG
104.      IF(ISW(26))430,435,430
105.    43C  JTAPE=108
106.      IREC1=2
107.      GO TO 440
108.    435  IF(ISW(33))460,440,460
109.    44C  WRITE(JTAPE,990)IREC1,ISTA,KEY,LAT,LATM,KNS,L0NG,L0M,
110.      1  KEW,J1,K1,J2,K2,J3,K3,J4,K4,J5,K5,J6,K6,J7,K7,J8,K8,
111.      2  IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,STHIK,CRVN,
112.      3  WGTN,AVWTN,CRVW,WGTW,AVWTW
113.      4  ,LTKEY,LGKEY,IAKEY
114.      RETURN
115.    46C  WRITE(JTAPE,991)IREC1,ISTA,KEY,LAT,LATM,KNS,L0NG,L0M,
116.      1  KEW,J1,K1,J2,K2,J3,K3,J4,K4,J5,K5,J6,K6,J7,K7,J8,K8,
117.      2  IMANT,NELEV,N1,N2,N3,N4,MET,IYR,IDESC,DINE,STHIK,CRVN,
118.      3  WGTN,AVWTN,CRVW,WGTW,AVWTW
119.      4  ,LTKEY,LGKEY,IAKEY

```

```
120•      RETURN
121•      900  CONTINUE
122•      WRITE(IIOUT,902)
123•      902  FORMAT('EOF FOUND, PROCESSING COMPLETED')
124•      KK=9
125•      RETURN
126•      END
```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
AVMTN	R	SCALR	*00039	V DUMMY	AVMTN	R	SCALR	*0003C	V DUMMY	CRVN	R	SCALR	*00037	V DUMMY	CRVN	R	SCALR	*00037	V DUMMY
CRVN	R	SCALR	*0003A	V DUMMY	DINE	R	SCALR	*00035	V DUMMY	FLBAT	R	SPR8G	INTRIN	1	FLBAT	R	SPR8G	INTRIN	1
IAKEY	I	SCALR	00004	V 1	IDESC	I	ARRAY	*00034	V DUMMY	IIN	I	SCALR	00001	V 1	IIN	I	SCALR	00001	V 1
IIOUT	I	SCALR	00002	V 1	IMANT	I	SCALR	*0002C	V DUMMY	IIRC1	I	SCALR	00006	V 1	IIRC1	I	SCALR	00006	V 1
ISTA	I	SCALR	*00022	V DUMMY	ISTAB	I	SCALR	000C3	V 1	ISW	I	SPR8G	EXTERN	1	ISW	I	SPR8G	EXTERN	1
ITAPE	I	SCALR	*0001F	V DUMMY	IYR	I	SCALR	*00033	V DUMMY	JTAPE	I	SCALR	*00020	V DUMMY	JTAPE	I	SCALR	*00020	V DUMMY
J1	I	SCALR	00007	V 1	J2	I	SCALR	000C9	V 1	J3	I	SCALR	0000B	V 1	J3	I	SCALR	0000B	V 1
J4	I	SCALR	0000D	V 1	J5	I	SCALR	000CF	V 1	J6	I	SCALR	00011	V 1	J6	I	SCALR	00011	V 1
J7	I	SCALR	00013	V 1	J8	I	SCALR	00015	V 1	KEW	I	SCALR	*00029	V DUMMY	KEW	I	SCALR	*00029	V DUMMY
KEY	I	SCALR	*00023	V DUMMY	KK	I	SCALR	*00021	V DUMMY	KNS	I	SCALR	*00026	V DUMMY	KNS	I	SCALR	*00026	V DUMMY
K1	I	SCALR	00008	V 1	K2	I	SCALR	000CA	V 1	K3	I	SCALR	0000C	V 1	K3	I	SCALR	0000C	V 1
K4	I	SCALR	0000E	V 1	K5	I	SCALR	00010	V 1	K6	I	SCALR	00012	V 1	K6	I	SCALR	00012	V 1
K7	I	SCALR	00014	V 1	K8	I	SCALR	00016	V 1	LAT	I	SCALR	*00024	V DUMMY	LAT	I	SCALR	*00024	V DUMMY
LATH	I	SCALR	*00025	V DUMMY	LGKEY	I	SCALR	0001E	V 1	LGM	I	SCALR	*00028	V DUMMY	LGM	I	SCALR	*00028	V DUMMY
LBNG	I	SCALR	*00027	V DUMMY	LTKEY	I	SCALR	00017	V 1	MET	I	SCALR	*00032	V DUMMY	MET	I	SCALR	*00032	V DUMMY
NAVIN	I	SPR8G	EXTERN	1	NELEV	I	SCALR	*0002D	V DUMMY	N1	I	SCALR	*0002E	V DUMMY	N1	I	SCALR	*0002E	V DUMMY
N2	I	SCALR	*0002F	V DUMMY	N3	I	SCALR	*00030	V DUMMY	N4	I	SCALR	*00031	V DUMMY	N4	I	SCALR	*00031	V DUMMY
PIN8T	R	SCALR	00000	V 1	PIN8T	R	SPR8G	000C0	P 1	PLAT	R	SCALR	0001C	V 1	PLAT	R	SCALR	0001C	V 1
PL8NG	R	SCALR	0001D	V 1	RADEG	R	SCALR	00005	V 1	RL8NG	R	SCALR	0001A	V 1	RL8NG	R	SCALR	0001A	V 1
PL8TH	R	SCALR	00018	V 1	RL8M	R	SCALR	00019	V 1	RL8NG	R	SCALR	0001B	V 1	RL8NG	R	SCALR	0001B	V 1
STHIK	R	SCALR	*00036	V DUMMY	THICK	R	ARRAY	*0002B	V DUMMY	VEL	R	ARRAY	*0002A	V DUMMY	VEL	R	ARRAY	*0002A	V DUMMY
WGTH	R	SCALR	*00038	V DUMMY	WGTH	R	SCALR	*0003B	V DUMMY										

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
10	00067	15	000BC	18	0010C	22	00111
25	00130	98	00185	400	00027	410	00067
430	001AC	440	001B6	460	001EC	600	00035
900	00222	990	000A0	991	000F0		

## LOCAL VARIABLES (31 WORDS):

0000C	PIN8T	00001	IIN	00005	RADEG
00006	IIRC1	00007	J1	0000B	J3
0000C	K3	0000C	J4	00011	J6
00012	K6	00013	J7	00017	LTKEY
00018	RL8TH	00019	RL8M	0001D	PL8NG
0001E	LGKEY				

BLANK COMMON (0 WORDS)

ENTRY POINTS:

0000C PIN8T

## INTRINSIC SUBPROGRAMS USED:

FLBAT

```

1.      SUBROUTINE PLANET(KK,RKM,PMASS,GM,PDENS)
2.      C      VERSION OF 23 MAR 1973, CHANGING SENSE SWITCH NUMBER
3.      C      VERSION OF 27 FEB 1973, INITIAL VERSION
4.      C      SUBROUTINE PLANET, RETURNS PARAMETER VALUES FOR
5.      C
6.      C      PLANET SELECTED BY SSW(38 & 39)
7.      C      SSW(38) = 0 FOR SPHERICAL EARTH
8.      C      = 1 FOR SPHERICAL MOON
9.      C      = 2 FOR SPHERICAL MARS
10.     C
11.     C      RKM = RADIUS IN KILOMETERS
12.     C      PMASS = PLANET MASS IN KILOGRAMS
13.     C      GM = GM IN DYNES/GRAMS SQUARED
14.     C      PDENS = MEAN DENSITY IN GRAMS/CC
15.     C
16.     DATA ISRT/0/
17.     IF(ISRT)10,5,10
18.     5  OUTPUT 'SUBROUTINE PLANET, VERSION OF 23 MAR, 1973'
19.     10  IF= ISW(38)
20.     IF(IP)20,20,30
21.     C  SPHERICAL EARTH
22.     20  RKM= 6371.2213
23.     PMASS= 5.983E+24
24.     GM= 3.990661E+20
25.     PDENS= 5.52
26.     GO TO 900
27.     30  CONTINUE
28.     GO TO(40,50)IP
29.     C  SPHERICAL MOON
30.     40  RKM=1738.0
31.     PMASS= 7.3554E+22
32.     C      REF= FIELDER,1961,P.245
33.     GM= 4.90605E+18
34.     PDENS= 3.34
35.     C  REF= FIELDER,1961,P. 245
36.     GO TO 900
37.     C  SPHERICAL MARS
38.     50  RKM=3394.0
39.     C  REF = LORELL ET AL., 1972, SCIENCE, V.175, P. 3179
40.     PMASS= 0.
41.     GM= 0.
42.     PDENS= 3.96
43.     GO TO 900
44.     900  CONTINUE
45.     KK= IP
46.     RETURN
47.     END

```

NAME	TYPE	CLASS	HEX L&C	DEC WORDS	NAME	TYPE	CLASS	HEX L&C	DEC WORDS	NAME	TYPE	CLASS	HEX L&C	DEC WORDS
GM	R	SCALR	*00006 V	DUMMY	IP	I	SCALR	00002 V	1	ISRT	I	SCALR	00001 V	1
ISM	I	SPR8G	EXTERN		KK	I	SCALR	*00003 V	DUMMY	PDENS	R	SCALR	*00007 V	DUMMY
PLANET	R	SPR8G	00000 F		PLANET	R	SCALR	00000 V	1	PMASS	R	SCALR	*00005 V	DUMMY
RKM	R	SCALR	*00004 V	DUMMY										

LABEL	HEX L&C	LABEL	HEX L&C	LABEL	HEX L&C
5	0000A	10	0001A	20	00020
900	00042			30	00029
				40	00030
				50	00039

## LOCAL VARIABLES (3 WORDS):

00000 PLANET 00001 ISRT 00002 IP

BLANK COMPON (0 WORDS)

ENTRY POINTS:

00000 PLANET

EXTERNAL SUBPROGRAMS REQUIRED:

ISM F:108 9ENDI9L 9PRINT 9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	69	00045
CONSTANTS:	11	00008
LOCAL VARIABLES:	3	00003
TEMPS:	6	00006
TOTAL PROGRAM:	89	00059

```

1. SUBROUTINE PLOTA(NB,A,N,M,NL,NS,ISCALE,IZERO,TMIN,TMAX)
2. C CATALOG NUMBER 890004 PLOT
3. C NOTE THAT THE NAME HAS BEEN CHANGED FROM PLOT TO PLOTA
4. C TO AVOID CONFLICT WITH THE CALCOMP SUBROUTINES
5. C NB= PLOT NUMBER-- THIS APPEARS AT THE START OF THE PLOT
6. C A ---- THE ARRAY IN WHICH THE INDEPENDANT VARIABLE AND THE
7. C DEPENDANT VARIABLES ARE STORED COLUMN WISE. IF THE ARRAY IS DIME(>23):
8. C DEPENDANT VARIABLES ARE STORED COLUMN WISE. IF THE ARRAY IS DI
9. C DIMENSIONED A(N,M) IN THE CALLING PROGRAM THE INDEPENDANT VARIABLE
10. C IS IN A(1,1) THRU A(N,1), THE FIRST VARIABLE TO BE PLOTTED IS
11. C STORED IN A(1,2) THRU A(N,2) AND SO ON.
12. C NL --- THE NUMBER OF POINTS IN EACH COLUMN OF THE ARRAY THAT
13. C WE WISH TO PLOT.
14. C NS =1 TO REORDER THE ARRAY SO THAT THE VALUES OF THE INDEPENDANT
15. C VARIABLE / INCRE
16. C VARIABLE ARE STORED IN INCREASING ORDER.
17. C =1 THE INDEPENDANT VARIABLE IS ALREADY STORED IN INCREASING ORDE
18. C ORDER
19. C
20. C
21. C
22. C
23. C
24. C
25. C ISCALE =0 SCALE THE ARRAY
26. C ISCALE =1 DO NOT SCALE THE ARRAY, USE THE LIMITS TMAX,TMIN
27. C AND IF IT IS GREATER THAN TMAX OR TMIN PUT THE VARIABLE AT THE
28. C EDGE
29. C IZERO =0 DO NOT PUT IN ZERO LINE, =1 PUT IN ZERO LINE
30. C TMAX,TMIN MAXIMUM AND MINIMUM VALUES PLOTTED IF ISCALE=1
31. C DIMENSION OUT(103),YPR(11),ANG(9),A(1)
32. C DATA BLANK,ANG/' ','1','2','3','4','5','6','7','8','9'/
33. C REAL LINE
34. C DATA LINE/' '/
35. C 1 FORMAT(1H1,60X,7H CHART ,I3,/)
36. C 2 FORMAT(1H ,F11.4,4X,103A1)
37. C 3 FORMAT (' ')
38. C 7 FORMAT(1H--,16X,1*,10('-----*'))
39. C 8 FORMAT(1H0,9X,11F10.4)
40. C YMAX=TMAX
41. C YMIN=TMIN
42. C IOT=108
43. C NLL=NL
44. C IF(NS) 16, 16, 10
45. C 10 DO 15 I=1,N
46. C DO 14 J=1,M
47. C IF(A(I)=A(J)) 14, 14, 11
48. C 11 L=I-N
49. C LL=J-N
50. C DO 12 K=1,M
51. C L=L+N
52. C LL=LL+N
53. C F=A(L)
54. C A(L)=A(LL)
55. C 12 A(LL)=F
56. C 14 CONTINUE
57. C 15 CONTINUE
58. C 16 IF(NLL) 20, 18, 20
59. C 18 NLL=50

```

```

60.      20 WRITE(IOT,1)N8
61.      WRITE(IOT,7)
62.      XSCAL=(A(NLL)-A(1))/(FLOAT(NLL-1))
63.      IF(ISCAL.EG.1) GO TO 42
64.      YMIN=1.0E75
65.      YMAX=-1.0E75
66.      DO 40 MC=1,M-1
67.      M1=N*MC+1
68.      M2=N*MC+NLL
69.      DO 40 J=M1,M2
70.      IF(A(J)-YMIN) 28,26,26
71.      26 IF(A(J)-YMAX) 40,40,30
72.      28 YMIN=A(J)
73.      GO TO 40
74.      30 YMAX=A(J)
75.      40 CONTINUE
76.      42 CONTINUE
77.      YSCAL=(YMAX-YMIN)/100.0
78.      XB=A(1)
79.      L=1
80.      MY=M-1
81.      DO 80 I=1,NLL
82.      F=I-1
83.      XPR=A(L)
84.      50 DO 55 IX=1,102
85.      55 OUT(IX)=BLANK
86.      OUT(1)=LINE
87.      OUT(103)=LINE
88.      IF(IZERO.EG.0) GO TO 59
89.      JZ=-YMIN/YSCAL+2.
90.      IF(JZ>1 .AND. JZ<103) OUT(JZ)=LINE
91.      59 CONTINUE
92.      DO 60 J=1,MY
93.      LL=L+J*N
94.      JP=((A(LL)-YMIN)/YSCAL)+2.0
95.      IF(JP>103) OUT(103)=ANG(J) ; GO TO 60
96.      IF(JP<1) OUT(1)=ANG(J) ; GO TO 60
97.      OUT(JP)=ANG(J)
98.      60 CONTINUE
99.      WRITE(IOT,2)XPR,(OUT(IZ),IZ=1,103)
100.     L=L+1
101.     GO TO 80
102.     70 WRITE(IOT,3)
103.     80 CONTINUE
104.     WRITE(IOT,7)
105.     YPR(1)=YMIN
106.     DO 90 KN=1,9
107.     90 YPR(KN+1)=YPR(KN)+YSCAL*10.0
108.     YPR(11)=YMAX
109.     WRITE(IOT,8)(YPR(IP),IP=1,11)
110.     78 FORMAT(10(E11.6,1X))
111.     RETURN
112.     END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
A	R	ARRAY	*00097	V DUMMY	ANG	R	ARRAY	00073	V	BLANK	R	SCALR	0007C	V
F	R	SCALR	00087	V	FLGAT	R	SPRGG	INTRIN	V	I	SCALR	00082	V	
IOT	I	SCALR	00080	V	IP	I	SCALR	00095	V	ISCALE	I	SCALR	*0009C	V DUMMY
IX	I	SCALR	00090	V	IZ	I	SCALR	00093	V	IZER0	I	SCALR	*0009D	V DUMMY
J	I	SCALR	00083	V	JP	I	SCALR	00092	V	JZ	I	SCALR	00091	V
K	I	SCALR	00086	V	KN	I	SCALR	00094	V	L	I	SCALR	00084	V
LINE	R	SCALR	0007D	V	LL	I	SCALR	00085	V	M	I	SCALR	*00099	V DUMMY
MC	I	SCALR	00089	V	LY	I	SCALR	0008E	V	M1	I	SCALR	0008A	V
M2	I	SCALR	00088	V	N	I	SCALR	*00098	V DUMMY	NL	I	SCALR	*0009A	V DUMMY
NLL	I	SCALR	00081	V	NB	I	SCALR	*00096	V DUMMY	NS	I	SCALR	*0009B	V DUMMY
OUT	R	ARRAY	00001	V	PLGTA	R	SCALR	00000	V	PLGTA	R	SPRGG	00000	P
TPAX	R	SCALR	*0009F	V DUMMY	TMIN	R	SCALR	00000	V	XB	R	SCALR	0008D	V
XPR	R	SCALR	0008F	V	XSCAL	R	SCALR	00088	V	YMAX	R	SCALR	0007E	V
YMIN	R	SCALR	0007F	V	YPR	R	ARRAY	00068	V	YSCAL	R	SCALR	0008C	V

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
1	0000E	3	0001C	7	00021	8	0002A
11	0004C	14	0005C	15	00062	16	00066
20	0006A	28	00096	30	0009F	40	000A2
50	000C1	59	000D4	60	000FB	70	00113
80	00117						

## LOCAL VARIABLES (150 WORDS):

000C0	PLGTA	000C1	9UT	0007C	BLANK	0007D	LINE
0007E	YMAX	0007F	YMIN	00082	I	00083	J
00084	L	00085	LL	00088	XSCAL	00089	MC
0008A	M1	0008B	M2	0008E	MY	0008F	XPR
0009C	IX	00091	JZ	00094	KN	00095	IP

BLANK COMPON (0 WORDS)

ENTRY POINTS:

00000 PLGTA

INTRINSIC SUBPROGRAMS USED:

FLOAT

EXTERNAL SUBPROGRAMS REQUIRED:

F:102	F:104	F:106	F:108	9BCDWRIT	9ENDIBL	9IBDATA	9IT0R
9RT01	9SETUPN						

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	320	00140
CONSTANTS:	5	00005
LOCAL VARIABLES:	150	00096
TEMPS:	13	0000D
	-----	-----
TOTAL PROGRAM:	488	001E8

```

1.      SUBROUTINE RETBY
2.      A( I DATA, IEBC, IIN, IISUT, ITAPE, NUMPL, DATA, RLAT, RLONG, KEGHM, IAGAP, LCN
3.      BT, RADEG, DEGRA, KDEG2, IDEG2, FDEG2, RDEG2, RTOP, ITOP, RBOT, IBOT, RLEFT,
4.      C ILEFT, RRIGT, IRIGT, SLTK, SLGK, SINCH, SMP, FBOT, FTOP, FLEFT, FRIGT, NDEG,
5.      D SLAT, SLONG, BOTMP, XX, YY, INIT, XOLD, YOLD)
6.      C
7.      C      SUBROUTINE RETBY,  TO SET  CHART BOUNDARIES
8.      C
9.      C
10.     CS      WRITE (IISUT,10)
11.     CS 10  FORMAT('E BND 1 LINE')
12.     CS      READ(IIN,*)ITOP,IBOT,ILEFT,IRIGT
13.     CS      READ(IIN,5)ITOP,IBOT,ILEFT,IRIGT
14.     5  FORMAT (4I5)
15.     5  OUTPLT ITOP, IBOT, ILEFT, IRIGT
16.     FTOP=ITOP
17.     FBOT=IBOT
18.     FLEFT=ILEFT
19.     FRIGT=IRIGT
20.     RTOP=FTOP*DEGRA
21.     RBOT=FBOT*DEGRA
22.     RLEFT=FLEFT*DEGRA
23.     RRIGT=FRIGT*DEGRA
24.     KDEG2=(IRIGT-ILEFT)
25.     IDEG2=KDEG2
26.     FDEG2=IDEG2
27.     RDEG2=FDEG2*DEGRA
28.     SMP=SINCH/60.0
29.     RETURN
30.     END

```

NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS	NAME	TYPE	CLASS	HEX L8C	DEC WORDS
BTMP		UNUSE	*00026	V DUMMY	DATA		UNUSE	*00007	V DUMMY	DEGRA		R SCALR	*0000E	V DUMMY	IRIGT		I SCALR	*0001A	V DUMMY
F80T	R	SCALR	*0001F	V DUMMY	FDEG2		R SCALR	*00011	V DUMMY	FLEFT		R SCALR	*00021	V DUMMY	IRIGT		I SCALR	*0001A	V DUMMY
FRIGT	R	SCALR	*00022	V DUMMY	FT8P		R SCALR	*00020	V DUMMY	IAGAP		UNUSE	*0000B	V DUMMY	KDEG2		I SCALR	*0000F	V DUMMY
I80T	I	SCALR	*00016	V DUMMY	IDATA		UNUSE	*000C1	V DUMMY	IADG2		I SCALR	*00010	V DUMMY	NDEG		UNUSE	*00023	V DUMMY
IE8C	UNUSE	*00002	V DUMMY		IIN		I SCALR	*000C3	V DUMMY	I18UT		UNUSE	*00004	V DUMMY	R80T		R SCALR	*00015	V DUMMY
ILEFT	I	SCALR	*00018	V DUMMY	INIT		UNUSE	*000C29	V DUMMY	IRIGT		I SCALR	*0000F	V DUMMY	RETBY		UNUSE	*00009	V DUMMY
ITAPE	UNUSE	*00005	V DUMMY		IT8P		I SCALR	*00014	V DUMMY	KDEG2		I SCALR	*0000F	V DUMMY	RLONG		R SCALR	*0001D	V DUMMY
K8GHM	UNUSE	*0000A	V DUMMY		LCNT		UNUSE	*000CC	V DUMMY	NDEG		UNUSE	*00023	V DUMMY	SLONG		UNUSE	*00025	V DUMMY
N8MPL	UNUSE	*00006	V DUMMY		RADEG		UNUSE	*000CD	V DUMMY	R80T		R SCALR	*00015	V DUMMY	X8LD		UNUSE	*0002A	V DUMMY
RDEG2	R	SCALR	*00012	V DUMMY	RETRY		SPROG	00000	P	RETBY		R SCALR	*00000	V DUMMY	YY		UNUSE	*00028	V DUMMY
RLAT	UNUSE	*00008	V DUMMY		RLEFT		R SCALR	*00017	V DUMMY	RLONG		R SCALR	*0001D	V DUMMY					
RRIGHT	R	SCALR	*00019	V DUMMY	RT8P		UNUSE	*0001C	V DUMMY	SLONG		UNUSE	*00025	V DUMMY					
SLAT	UNUSE	*00024	V DUMMY		SLGK		R SCALR	*0001E	V DUMMY	X8LD		UNUSE	*0002A	V DUMMY					
SLTK	UNUSE	*0001B	V DUMMY		SMP					YY		UNUSE	*00028	V DUMMY					
XX	UNUSE	*00027	V DUMMY		Y8LD														

LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C	LABEL	HEX L8C
5	00036						

## LOCAL VARIABLES (1 WORD):

C0000 RETBY

BLANK COMPON (C WORDS)

ENTRY POINTS:

C0000 RETBY

## EXTERNAL SUBPROGRAMS REQUIRED:

F:101	F:103	F:105	F:108	9BCDREAD	918DATA	91T8R	9PRINT
9SETUPN							

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	117	C0C75
CONSTANTS:	1	C0C01
LOCAL VARIABLES:	1	C0C01
TEMPS:	44	C0C2C
TOTAL PROGRAM:	163	C0CA3

```
1.      SUBROUTINE RTDM2(RAD, IDEG, AMIN)
2.  C    SUBROUTINE RTDM2 TO CONVERT RADIANS (RAD) TO
3.  C    DEGREES (IDEG) AND MINUTES (AMIN)
4.  C    FOR AMIN WITH 2 DIGITS TO RIGHT OF DECIMAL
5.      B = RAD*57.29578
6.      A=ABS(B)
7.      A=A+C.00005
8.      A=SIGN(A,B)
9.      IDEG=A
10.     A = IDEG
11.     AMIN = (B-A)*60.0
12.     AMIN = SIGN(AMIN,RAD)
13.     RETURN
14.     END
```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	CCCC2 V	1	ABS	R	SPR8G	INTRIN	1	AMIN	R	SCALR	*00005 V	DUMMY
B	R	SCALR	CCCC1 V	1	IDEG	I	SCALR	*00004 V	DUMMY	RAD	R	SCALR	*00003 V	DUMMY
RTDM2	SPR8G	CCCC0 P			RTDM2	R	SCALR	CCCC0 V	1	SIGN	R	SPR8G	INTRIN	

## LOCAL VARIABLES (3 WORDS):

CCCCC RTDM2	CCCC1 B	000C2 A
-------------	---------	---------

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

CCCCC RTDM2
-------------

## INTRINSIC SUBPROGRAMS USED:

ABS	SIGN
-----	------

## EXTERNAL SUBPROGRAMS REQUIRED:

91T8R	9RT8I	9SETUPN
-------	-------	---------

## HIGHEST ERROR SEVERITY: C (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	35	00023
CONSTANTS:	3	00003
LOCAL VARIABLES:	3	00003
TEMPS:	4	00004
TOTAL PROGRAM:	45	0002D

```
1.      SUBROUTINE RT0DM(RAD, IDEG, AMIN)
2.  C    SUBROUTINE RT0DM TO CONVERT RADIAN (RAD) TO
3.  C    DEGREES (IDEG) AND MINUTES (AMIN)
4.  C    FOR AMIN WITH 0 DIGITS TO RIGHT OF DECIMAL
5.      B = RAD*57.29578
6.      A=ABS(B)
7.      A=A+0.005
8.      A=SIGN(A,B)
9.      IDEG=A
10.     A = IDEG
11.     AMIN = (B-A)*60.0
12.     AMIN = SIGN(AMIN,RAD)
13.     RETURN
14.     END
```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
A	R	SCALR	00002 V	1	ABS	R	SPR8G	INTRIN	1	AMIN	R	SCALR	*00005 V	DUMMY
B	R	SCALR	00001 V	1	IDEG	I	SCALR	*00004 V	DUMMY	RAD	R	SCALR	*00003 V	DUMMY
RT8DM	SPR8G	00000 P			RT8DM	R	SCALR	00000 V	1	SIGN	R	SPR8G	INTRIN	

## LOCAL VARIABLES (3 WORDS):

00000 RT8DM	00001 B	00002 A
-------------	---------	---------

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

00000 RT8DM
-------------

## INTRINSIC SUBPROGRAMS USED:

ABS	SIGN
-----	------

## EXTERNAL SUBPROGRAMS REQUIRED:

9110F	9RT8I	9SETUPN
-------	-------	---------

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	35	00023
CONSTANTS:	3	00003
LOCAL VARIABLES:	3	00003
TEMPS:	4	00004
TOTAL PROGRAM:	45	0002D

```

1.      SUBROUTINE SIMUL(N,A,X,EPS,INDIC,NRC,DETER)
2.      C
3.      C      WHEN INDIC IS NEGATIVE, SIMUL COMPUTES THE INVERSE OF THE N BY N
4.      C      MATRIX A IN PLACE. WHEN INDIC IS ZERO, SIMUL COMPUTES THE N
5.      C      SOLUTIONS X(1)...X(N) CORRESPONDING TO THE SET OF LINEAR
6.      C      EQUATIONS WITH AUGMENTED MATRIX OF COEFFICIENTS IN THE N BY
7.      C      N+1 ARRAY A AND IN ADDITION COMPUTES THE INVERSE OF THE
8.      C      COEFFICIENT MATRIX IN PLACE AS ABOVE. IF INDIC IS POSITIVE,
9.      C      THE SET OF LINEAR EQUATIONS IS SOLVED BUT THE INVERSE IS NOT
10.     C      COMPUTED IN PLACE. THE GAUSS-JORDAN COMPLETE ELIMINATION METHOD
11.     C      IS EMPLOYED WITH THE MAXIMUM PIVOT STRATEGY. ROW AND COLUMN
12.     C      SUBSCRIPTS OF SUCCESSIVE PIVOT ELEMENTS ARE SAVED IN ORDER IN
13.     C      THE IROW AND JCOL ARRAYS RESPECTIVELY. K IS THE PIVOT COUNTER,
14.     C      PIVOT THE ALGEBRAIC VALUE OF THE PIVOT ELEMENT, MAX
15.     C      THE NUMBER OF COLUMNS IN A AND DETER THE DETERMINANT OF THE
16.     C      COEFFICIENT MATRIX. THE SOLUTIONS ARE COMPUTED IN THE (N+1)TH
17.     C      COLUMN OF A AND THEN UNSCRAMBLED AND PUT IN PROPER ORDER IN
18.     C      X(1)...X(N) USING THE PIVOT SUBSCRIPT INFORMATION AVAILABLE
19.     C      IN THE IROW AND JCOL ARRAYS. THE SIGN OF THE DETERMINANT IS
20.     C      ADJUSTED, IF NECESSARY, BY DETERMINING IF AN ODD OR EVEN NUMBER
21.     C      OF PAIRWISE INTERCHANGES IS REQUIRED TO PUT THE ELEMENTS OF THE
22.     C      JORD ARRAY IN ASCENDING SEQUENCE WHERE JORD(IROW(I)) = JCOL(I).
23.     C      IF THE INVERSE IS REQUIRED, IT IS UNSCRAMBLED IN PLACE USING
24.     C      Y(1)...Y(N) AS TEMPORARY STORAGE. THE VALUE OF THE DETERMINANT
25.     C      IS RETURNED AS THE VALUE OF THE FUNCTION. SHOULD THE POTENTIAL
26.     C      PIVOT OF LARGEST MAGNITUDE BE SMALLER IN MAGNITUDE THAN EPS,
27.     C      THE MATRIX IS CONSIDERED TO BE SINGULAR AND A TRUE ZERO IS
28.     C      RETURNED AS THE VALUE OF THE FUNCTION.
29.     C
30.     C      REFERENCE: CARNAHAN, LUTHER AND WILKES (1969)
31.     C      APPLIED NUMERICAL METHODS. WILEY, NEW YORK.
32.     C
33.     C      CONVERTED TO XDS FORTRAN IV-H BY H. PERKINS, APRIL, 1970.
34.     C
35.     C      DIMENSION IROW(50), JCOL(50), JORD(50), Y(50), A(NRC,NRC), X(N)
36.     C
37.     C      MAX = N
38.     C      IF ( INDIC.GE.0 ) MAX = N + 1
39.     C
40.     C      .....IS N LARGER THAN 50 .....
41.     C      IF ( N.LE.50 ) GO TO 5
42.     C      WRITE(108,200)
43.     C      SIMUL = 0.
44.     C      RETURN
45.     C
46.     C      ..... BEGIN ELIMINATION PROCEDURE .....
47.     C      5 DETER = 1.
48.     C      DO 18 K = 1, N
49.     C      KM1 = K - 1
50.     C
51.     C      ..... SEARCH FOR THE PIVOT ELEMENT .....
52.     C      PIVOT = 0.
53.     C      DO 11 I = 1, N
54.     C      DO 11 J = 1, N
55.     C      ..... SCAN IROW AND JCOL ARRAYS FOR INVALID PIVOT SUBSCRIPTS .....
56.     C      IF ( K.EQ.1 ) GO TO 9
57.     C      DO 8 ISCAN = 1, KM1
58.     C      DO 8 JSCAN = 1, KM1
59.     C      IF ( I.EQ.IROW(ISCAN) ) GO TO 11

```

```

60.      8 IF ( J.EG.JCOL(JSCAN) ) GO TO 11          SIMU0060
61.      9 IF ( DABS(A(I,J)).LE.DABS(PIVOT) ) GO TO 11  SIMU0061
62.      PIVOT = A(I,J)                               SIMU0062
63.      IRBK(K) = I                                    SIMU0063
64.      JCOL(K) = J                                    SIMU0064
65.      11 CONTINUE                                    SIMU0065
66.      C                                             SIMU0066
67.      C ..... INSURE THAT SELECTED PIVOT IS LARGER THAN EPS ..... SIMU0067
68.      IF ( DABS(PIVOT).GT.EPS ) GO TO 13           SIMU0068
69.      SIMUL = 0.                                     SIMU0069
70.      RETURN                                         SIMU0070
71.      C                                             SIMU0071
72.      C ..... UPDATE THE DETERMINANT VALUE ..... SIMU0072
73.      13 IRBK = IRBK(K)                             SIMU0073
74.      JCOL = JCOL(K)                               SIMU0074
75.      DETER = DETER*PIVOT                           SIMU0075
76.      C                                             SIMU0076
77.      C ..... NORMALIZE PIVOT ROW ELEMENTS ..... SIMU0077
78.      DO 14 J = 1, MAX                             SIMU0078
79.      14 A(IRBK,J) = A(IRBK,J)/PIVOT                SIMU0079
80.      C                                             SIMU0080
81.      C ..... CARRY OUT ELIMINATION AND DEVELOP INVERSE ..... SIMU0081
82.      A(IRBK,JCOL) = 1./PIVOT                      SIMU0082
83.      DO 18 I = 1, N                                SIMU0083
84.      AIJCK = A(I,JCOL)                             SIMU0084
85.      IF ( I.EG.IRBK ) GO TO 18                    SIMU0085
86.      A(I,JCOL) = - AIJCK/PIVOT                     SIMU0086
87.      DO 17 J = 1, MAX                             SIMU0087
88.      17 IF ( J.NE.JCOL ) A(I,J) = A(I,J) - AIJCK*A(IRBK,J) SIMU0088
89.      18 CONTINUE                                    SIMU0089
90.      C                                             SIMU0090
91.      C ..... ORDER SOLUTION VALUES (IF ANY) AND CREATE JORD ARRAY ..... SIMU0091
92.      DO 20 I = 1, N                                SIMU0092
93.      IRBWI = IRBK(I)                               SIMU0093
94.      JCOLI = JCOL(I)                               SIMU0094
95.      JORD(IRBWI) = JCOLI                           SIMU0095
96.      20 IF ( INDIC.GE.0 ) X(JCOLI) = A(IRBWI,MAX)   SIMU0096
97.      C                                             SIMU0097
98.      C ..... ADJUST SIGN OF DETERMINANT ..... SIMU0098
99.      IF(N.EG.1) GO TO 24                           SIMU0099
100.     INTCH = 0                                       SIMU0100
101.     NM1 = N - 1                                     SIMU0101
102.     DO 22 I = 1, NM1                               SIMU0102
103.     IP1 = I + 1                                     SIMU0103
104.     DO 22 J = IP1, N                               SIMU0104
105.     IF ( JORD(J).GE.JORD(I) ) GO TO 22             SIMU0105
106.     JTEMP = JORD(J)                                SIMU0106
107.     JORD(J) = JORD(I)                              SIMU0107
108.     JORD(I) = JTEMP                                SIMU0108
109.     INTCH = INTCH + 1                               SIMU0109
110.     22 CONTINUE                                    SIMU0110
111.     IF ( INTCH/2*2.NE.INTCH ) DETER = - DETER     SIMU0111
112.     C                                             SIMU0112
113.     C ..... IF INDIC IS POSITIVE RETURN WITH RESULTS ..... SIMU0113
114.     24 IF ( INDIC.LE.0 ) GO TO 26                 SIMU0114
115.     SIMUL = DETER                                   SIMU0115
116.     RETURN                                         SIMU0116
117.     C                                             SIMU0117
118.     C ..... IF INDIC IS NEGATIVE OR ZERO, UNSCRAMBLE THE INVERSE SIMU0118
119.     C FIRST BY ROWS .....

```

120.	26	DO 28 J = 1, N	SIMU0119
121.		DO 27 I = 1, N	SIMU0120
122.		IR0W1 = IR0W(I)	SIMU0121
123.		J0BL1 = J0BL(I)	SIMU0122
124.	27	Y(J0BL1) = A(IR0W1,J)	SIMU0123
125.		DO 28 I = 1, N	SIMU0124
126.	28	A(I,J) = Y(I)	SIMU0125
127.	C	..... THEN BY COLUMNS .....	SIMU0126
128.		DO 30 I = 1, N	SIMU0127
129.		DO 29 J = 1, N	SIMU0128
130.		IR0WJ = IR0W(J)	SIMU0129
131.		J0BLJ = J0BL(J)	SIMU0130
132.	29	Y(IR0WJ) = A(I,J0BLJ)	SIMU0131
133.		DO 30 J = 1, N	SIMU0132
134.	30	A(I,J) = Y(J)	SIMU0133
135.	C		SIMU0134
136.	C	..... RETURN FOR INDIC NEGATIVE OR ZERO .....	SIMU0135
137.		SIMUL = DETER	SIMU0136
138.		RETURN	SIMU0137
139.	C		SIMU0138
140.	C	..... FORMAT FOR OUTPUT STATEMENT .....	SIMU0139
141.	200	FORMAT( 10F0N T00 BIG )	SIMU0140
142.	C		SIMU0141
143.		END	SIMU0142

NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS
A	R	ARRAY	*000D	V DUMMY	ALJCK	R	SCALR	000D3	V DUMMY	DABS	D	SPR6G	INTRIN	1	DABS	D	SPR6G	INTRIN	1
DETER	R	SCALR	*000E2	V DUMMY	EPS	R	SCALR	*000CF	V DUMMY	I	I	SCALR	000CD	V	I	I	SCALR	000CD	V
INDIC	R	SCALR	*000E0	V DUMMY	INTCH	I	SCALR	000C6	V	IP1	I	SCALR	000D8	V	IP1	I	SCALR	000D8	V
IR6WK	I	ARRAY	000C1	V 50	IR6WI	I	SCALR	000C4	V	IR6WJ	I	SCALR	000DA	V	IR6WJ	I	SCALR	000DA	V
IR6WK	I	SCALR	000C1	V 1	ISCAN	I	SCALR	000CF	V	J	I	SCALR	000CE	V	J	I	SCALR	000CE	V
JCBL	I	SCALR	00033	V 50	JCBLI	I	SCALR	000C5	V	JCBLJ	I	SCALR	000DB	V	JCBLJ	I	SCALR	000DB	V
JCBLK	I	SCALR	000D2	V 1	J6RD	I	ARRAY	000C5	V	JSCAN	I	SCALR	000D0	V	JSCAN	I	SCALR	000D0	V
JTEMP	I	SCALR	000C9	V 1	K	I	SCALR	000CA	V	KM1	I	SCALR	000CB	V	KM1	I	SCALR	000CB	V
JMAX	I	SCALR	000C9	V 1	N	R	PIV3T	*000CC	V DUMMY	NM1	I	SCALR	000D7	V	NM1	I	SCALR	000D7	V
NRC	I	SCALR	*000E1	P DUMMY	PIV3T	R	SCALR	000CC	V DUMMY	SIMUL	R	SCALR	000D0	V	SIMUL	R	SCALR	000D0	V
SIMUL	R	SPR6G	000CC	P	X	R	ARRAY	*000CE	V DUMMY	Y	R	ARRAY	00097	V	Y	R	ARRAY	00097	V

LABEL	HEX L6C	LABEL	HEX L6C	LABEL	HEX L6C	LABEL	HEX L6C
5	00020	9	00043	11	00056	13	0006C
17	00099	20	000C1	22	000E7	24	000F9
27	00106	29	00127	30	00133	200	00143

## LOCAL VARIABLES (220 WORDS):

000C	SIMUL	000C1	IR6WK
000CA	K	000CB	KM1
000CC	JSCAN	000CD	IR6WK
000C6	INTCH	000C7	NM1

BLANK COMMON (0 WORDS)

ENTRY PRINTS:

000C SIMUL

INTRINSIC SUBPROGRAMS USED:

DABS

EXTERNAL SUBPROGRAMS REQUIRED:

F:108 SEND10L SPRINT 9SETUPN

LABEL	HEX L6C	LABEL	HEX L6C
14	00076	000C9	MAX
26	000FE	000CF	ISCAN
		000D5	JCBLI
		000DB	JCBLJ

LABEL	HEX L6C	LABEL	HEX L6C
00097	Y	000CE	J
000D4	IR6WI	000D4	IR6WI
000DA	IR6WJ	000DA	IR6WJ

LABEL	HEX L6C	LABEL	HEX L6C
00065	J6RD	000C5	J6RD
000C3	I	000C3	I
000C3	AIJCK	000C3	AIJCK
000C9	JTEMP	000C9	JTEMP

LABEL	HEX L6C	LABEL	HEX L6C
00033	JCBL	00033	JCBL
000CC	PIV6T	000CC	PIV6T
000D2	JCBLK	000D2	JCBLK
000D8	IP1	000D8	IP1

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	329	CC149
CONSTANTS:	2	CC002
LOCAL VARIABLES:	220	CC0DC
TEMPS:	12	CC00C
	-----	-----
TOTAL PROGRAM:	563	CC233

```

1.      SUBROUTINE SPLBT(ISTA,RLAT,RLONG,VEL,THICK,VELW,WATTK,VMANT,XX,YY,
2.      1      YFAC,ZHT,HGT,AC,AS,ANGB,BDIST)
3.      C  SUBROUTINE SPLBT,  PLOTS SEISMIC REFRACTION PROFILE DATA
4.      C
5.      DIMENSION      VEL(8),THICK(8)
6.      DATA RADEG/57.29578/
7.      NCT=8
8.      C  LABLING COLUMN
9.      AISTA = ISTA
10.     XN = -0.10 * ZHT
11.     YN = 0.40
12.     XP = (XN*AC)+(YN*AS)
13.     YP = -1.0*(XN*AS) + (YN*AC)
14.     XT = XX + XP
15.     YT = YY+YP
16.     CALL NUMBER(XT,YT,HGT,AISTA,ANGB,-1)
17.     CALL PLBT(XX,YY,3)
18.     IF(ISH(5))490,465,490
19. 465   XN = -0.10
20.       YN = 0.40+(0.12*ZHT)
21.       XP = (XN*AC)+(YN*AS)
22.       YP = -1.0*(XN*AS) + (YN*AC)
23.       XT = XX + XP
24.       YT = YY+YP
25.       DLONG = RLONG*RADEG
26.       CALL NUMBER(XT,YT,HGT,DLONG,ANGB,2)
27.       XN = -0.10
28.       YN = 0.40+(0.24*ZHT)
29.       XP = (XN*AC)+(YN*AS)
30.       YP = -1.0*(XN*AS) + (YN*AC)
31.       XT = XX + XP
32.       YT = YY+YP
33.       DLAT = RLAT*RADEG
34.       CALL NUMBER(XT,YT,HGT,DLAT,ANGB,2)
35.       CALL PLBT(XX,YY,3)
36.      C  BEGIN PLOTTING COLUMN
37. 490   LIND = 0
38.       MAX = NCT+1
39.       IF(VELW=0.001)510,510,500
40. 500   CVEL = VELW
41.       DOWN = -WATTK/YFAC
42.       GO TO 100
43. 510   LIND = LIND + 1
44.       IF(LIND=MAX)512,530,530
45. 512   IF(VEL(LIND)-0.001)510,510,520
46. 520   CVEL = VEL(LIND)
47.       IF(THICK(LIND)-0.001)522,522,524
48. 522   IF(CVEL=7.0)524,523,523
49. 523   DOWN=-(BDIST * 2.0)
50.       GO TO 100
51. 524   DOWN = -THICK(LIND)/YFAC
52.       GO TO 100
53. 530   IF(VMANT=0.001)550,550,540
54. 540   CVEL = VMANT
55.       DOWN = -(BDIST * 2.0)
56.       VMANT = 0.0
57.       GO TO 100
58.      C  COMPLETED PLOTTING THIS COLUMN
59. 550   GO TO 750

```

```

60. CC
61. C PLOTTING COLUMN INSTRUCTIONS
62. CC
63. C WRITING TICK LINE AT ANGLE (ANGB)
64. 100 XP = (BDIST*AC)
65. YP = -1.0*(BDIST * AS)
66. XT = XX+(ABS(XP))
67. YT = YY+YP
68. CALL PLOT(XT,YT,2)
69. CALL PLOT(XX,YY,3)
70. C WRITING VELOCITY VALUE
71. 110 XN = -0.25 * ZHT
72. YN = -0.09 * ZHT
73. XF = (XN*AC)+(YN*AS)
74. YP = -1.0*(XN*AS)+(YN*AC)
75. XT=XX+XP
76. YT=YY+YP
77. CALL NUMBER(XT,YT,HGT,CVEL,ANGB,1)
78. CALL PLOT(XX,YY,3)
79. C CONTINUE PLOTTING DOWNWARD LINE
80. 120 XF= DOWN *AS
81. YF= DOWN *AC
82. XX= XX+XF
83. YY= YY+YP
84. CALL PLOT(XX,YY,2)
85. GO TO 510
86. 750 RETURN
87. END

```

[illegible]

00000	SFL0T	00001	RADEG	00002	NCT	00003	AISTA	00004	XN	00005	YN
00000	XF	00007	YF	00008	XT	00009	YT	0000A	DL0NG	0000B	DLAT
00000	LIND	0000D	MAX	0000E	CVEL	0000F	DOWN				

BLANK COMMON (C WORDS)

## ENTRY POINTS:

COCOC SPL0T

INTRINSIC SUBPROGRAMS USED:

## ABS

EXTERNAL SUBPROGRAMS REQUIRED:

ISK	NUMBER	PLOT	SITOR	9SETUFI
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28
29	29	29	29	29
30	30	30	30	30
31	31	31	31	31
32	32	32	32	32
33	33	33	33	33
34	34	34	34	34
35	35	35	35	35
36	36	36	36	36
37	37	37	37	37
38	38	38	38	38
39	39	39	39	39
40	40	40	40	40
41	41	41	41	41
42	42	42	42	42
43	43	43	43	43
44	44	44	44	44
45	45	45	45	45
46	46	46	46	46
47	47	47	47	47
48	48	48	48	48
49	49	49	49	49
50	50	50	50	50
51	51	51	51	51
52	52	52	52	52
53	53	53	53	53
54	54	54	54	54
55	55	55	55	55
56	56	56	56	56
57	57	57	57	57
58	58	58	58	58
59	59	59	59	59
60	60	60	60	60
61	61	61	61	61
62	62	62	62	62
63	63	63	63	63
64	64	64	64	64
65	65	65	65	65
66	66	66	66	66
67	67	67	67	67
68	68	68	68	68
69	69	69	69	69
70	70	70	70	70
71	71	71	71	71
72	72	72	72	72
73	73	73	73	73
74	74	74	74	74
75	75	75	75	75
76	76	76	76	76
77	77	77	77	77
78	78	78	78	78
79	79	79	79	79
80	80	80	80	80
81	81	81	81	81
82	82	82	82	82
83	83	83	83	83
84	84	84	84	84
85	85	85	85	85
86	86	86	86	86
87	87	87	87	8

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	299	0012B
CONSTANTS:	16	00010
LOCAL VARIABLES:	16	00010
TEMPS:	19	00013
	-----	-----
TOTAL PROGRAM:	350	0015E

```

1.      SUBROUTINE SPOT(XX,YY)
2.      C
3.      C  SUBROUTINE SPOT PLOTS A CIRCLE AT DATA POINT
4.      C
5.      C  USES CALCOMP SUBROUTINES
6.      C
7.      ST = XX + 0.02
8.      TT = YY - 0.01
9.      CALL PLOT(ST,TT,3)
10.     YT = YY + 0.01
11.     CALL PLOT(ST,YT,2)
12.     XT = XX + 0.01
13.     YT = YY + 0.02
14.     CALL PLOT(XT,YT,2)
15.     XT = XX - 0.01
16.     CALL PLOT(XT,YT,2)
17.     XT = XX - 0.02
18.     YT = YY + 0.01
19.     CALL PLOT(XT,YT,2)
20.     YT = YY - 0.01
21.     CALL PLOT(XT,YT,2)
22.     XT = XX - 0.01
23.     YT = YY - 0.02
24.     CALL PLOT(XT,YT,2)
25.     XT = XX + 0.01
26.     CALL PLOT(XT,YT,2)
27.     CALL PLOT(ST,TT,2)
28.     CALL PLOT(XX,YY,3)
29.     RETURN
30.     END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
PL0T	----	----	----	----	SP0T	----	----	----	----	SP0T	----	----	----	----
ST	R	SCALR	00001 V	1	TT	R	SCALR	00002 V	1	XT	R	SCALR	00004 V	1
XX	R	SCALR	*00005 V	DUMMY	YT	R	SCALR	00003 V	1	YY	R	SCALR	*00006 V	DUMMY

## LOCAL VARIABLES (5 WORDS):

00000 SP0T	00001 ST	00002 TT	00003 YT	00004 XT
------------	----------	----------	----------	----------

## BLANK COMPON (0 WORDS)

## ENTRY POINTS:

00000 SP0T

## EXTERNAL SUBPROGRAMS REQUIRED:

PL0T 9SETUP2

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	91	0005B
CONSTANTS:	6	00006
LOCAL VARIABLES:	5	00005
TEMPS:	3	00003
TOTAL PROGRAM:	105	00069

AD-A035 454

WHOI-77-2  
GRAVITY DATA PROCESSING PROGRAMS.(U)

FEB 77

CARL BOWIN

UNCLASSIFIED WOODS HOLE OCEANOGRAPHIC INSTITUTION, MA

6 OF 6  
AD-A  
035 454



END  
DATE  
FILMED  
4-8-77  
NTIS

```

1.      SUBROUTINE SPOT2(XX,YY)
2.      C      USES CALCOMP SUBROUTINES
3.      C
4.      C      MODIFIED TO MAKE SSW(4) DEFAULT NO MARKING      3 OCT 72
5.      C      VERSION OF 20 AUGUST 1971, CHECKS ISW(4) TO DEFINE SYMBOL ANNOTATED
6.      C
7.      C      SSW(4) = C FOR SUPPRESSING PLOTTING OF ANY SPOT
8.      C      = 1 FOR PLOTTING A CIRCLE AT DATA POINT
9.      C      = 2 FOR PLOTTING A DOT AT DATA POINT
10.     C
11.     NSW=ISW(4)
12.     C INCREMENT INDEX BY ONE TO PERMIT USE OF GO TO STATEMENT
13.     NSW=NSW+1
14.     GO TO (999,100,200,300,400,500,600,700,800,900)NSW
15.     C
16.     C      TO PLOT A CIRCLE AROUND DATA POINT
17.     100  ST = XX + 0.02
18.         TT = YY - 0.01
19.         CALL PLOT(ST,TT,3)
20.         YT = YY + 0.01
21.         CALL PLOT(ST,YT,2)
22.         XT = XX + 0.01
23.         YT = YY + 0.02
24.         CALL PLOT(XT,YT,2)
25.         XT = XX - 0.01
26.         CALL PLOT(XT,YT,2)
27.         XT = XX - 0.02
28.         YT = YY + 0.01
29.         CALL PLOT(XT,YT,2)
30.         YT = YY - 0.01
31.         CALL PLOT(XT,YT,2)
32.         XT = XX - 0.01
33.         YT = YY - 0.02
34.         CALL PLOT(XT,YT,2)
35.         XT = XX + 0.01
36.         CALL PLOT(XT,YT,2)
37.         CALL PLOT(ST,TT,2)
38.         CALL PLOT(XX,YY,3)
39.     999  RETURN
40.     C
41.     C      TO PLOT ONLY A DOT BY LOWERING AND RAISING PEN
42.     200  CALL PLOT(XX,YY,2)
43.         CALL PLOT(XX,YY,3)
44.     C
45.     C      OTHER OPTIONS TO BE IMPLEMENTED
46.     300  RETURN
47.     400  RETURN
48.     500  RETURN
49.     600  RETURN
50.     700  RETURN
51.     800  RETURN
52.     900  RETURN
53.     END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
ISM	I	SPR8G	EXTERNAL	1	NSH	I	SCALR	CCCC1 V	1	PL8T			EXTERNAL	1
SP8T2	R	SCALR	CCCC V	1	SP8T2	R	SPR8G	CCCC P	1	ST	R	SCALR	00002 V	1
TT	R	SCALR	CCCC V	1	XT	R	SCALR	CCCC V	1	XX	R	SCALR	*00006 V	DUMMY
YT	R	SCALR	CCCC4 V	1	YY	R	SCALR	*00007 V	DUMMY					

HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL
100 CCG18	200 C006F	300 C0079	400 C007A	500 0007B	600 0007C
700 CCG7D	800 C007E	900 C007F	999 C006E		

## LOCAL VARIABLES (6 WORDS):

CCCC SP8T2	CCCC1 NSH	CCCC2 ST	CCCC3 TT	00004 YT	00005 XT
------------	-----------	----------	----------	----------	----------

## BLANK COMPON (0 WORDS)

## ENTRY POINTS:

CCCC SP8T2

## EXTERNAL SUBPROGRAMS REQUIRED:

ISM PLOT 9SETUP2

## HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE:	DEC WORDS	HEX WORDS
CONSTANTS:	128	C0080
LOCAL VARIABLES:	7	C0007
TEMP:	6	C0006
	3	C0003
TOTAL PROGRAM:	144	C0090

```

1. SUBROUTINE TIDAL(RLAT,RLONG,IY,ID,IH,IM,CLS,HONK,DTD)
2. C WRITTEN BY C. GANTAR ACCORDING TO LONGMAN AND USCGS FORMULAE
3. C ADAPTED FOR SIGMA=7 ALG. 69 BY J. WOLFE
4. C THE OUT PUT OF THE PROGRAM HAS BEEN CHECKED WITH THE G. PRSS.
5. C BOOK FOR THE YEAR 69 WEIRD RESULTS BUT ARE IN GENERAL AGREEMENT
6. C
7. C RLAT(+ 1P NORTH),RLONG(+ 1F EAST)=GEOGR COORD. IN RADIANS
8. C IY=YEAR=1900 ID=DAY,PROGRESSIVE OF THE YEAR
9. C IH,IM=HOUR, MINUTES (GMT)
10. C CLS=TIDAL CORRECTION(MGAL)
11. C HONK=HONKASAL0 TERM(MGAL) TO BE ADDED TO CLS
12. C DOUBLE PRECISION DIY,DID,DTD,DBIS,CENT,DS,S,DH,H,DP,DN,DP1,P1,P,
13. C 10MEGA,SP,DSP,SHP,DSH,S1,SENBM,CBSBM,SENNE,CBSNE,SENNU,SENAL,CBSAL,
14. C 2DE1,L,L1,CH1,CH11,CSZ,CST,DMUN,RH0,FC1,PC2,PC3,DSUN,AI,SENI,FNU
15. C DOUBLE PRECISION N
16. C DOUBLE PRECISION RLAT,RLONG,CASBL
17. C DOUBLE PRECISION ARCS,ARSIN
18. C DOUBLE PRECISION DYM,DIBIS
19. C IMPLICIT REAL (N,L)
20. C X=FLOAT(IY)/4.+1
21. C X1=X-FLOAT(INT(X))
22. C IBIS=INT(X)
23. C IF(X1.LT.C*2)IBIS=IBIS-1
24. C DIY=IY
25. C DID=ID-1
26. C DIH=IH
27. C DYM=IM
28. C DIBIS=IBIS
29. C DTD=DIY*365.D0+DIBIS*0.5DC+DID+.041666667D0*DIH+.000694444D0*DYM
30. C CENT=DTD/36525.D0
31. C DS=4.720009D0+8399.709275D0*CENT+.000035D0*CENT**2
32. C S=DMOD(DS,6.28318530718D0)
33. C DH=4.881628D0+628.331951D0*CENT+.000005D0*CENT**2
34. C H=DMOD(DH,6.28318530718D0)
35. C DP=5.835152D0+71.018041D0*CENT+.00018D0*CENT**2
36. C P=DMOD(DP,6.28318530718D0)
37. C DN=4.523603D0+33.757146D0*CENT+.000036D0*CENT**2
38. C N=DMOD(DN,6.28318530718D0)
39. C DP1=4.908229D0+0.030003D0*CENT+.000008D0*CENT**2
40. C P1=DMOD(DP1,6.28318530718D0)
41. C 0MEGA=0.409320D0+0.000227D0*CENT
42. C SP=S-P
43. C DSP=2.D0*SP
44. C SHP=S-2.D0*H+P
45. C DSH=2.D0*(S-H)
46. C S1=S+.1098D0*DSIN(SP)+.0037675D0*DSIN(DSP)+.0154001D0*DSIN(SHP)+.0
47. C 1076939D0*DSIN(DSH)
48. C SENBM=DSIN(0MEGA)
49. C CBSBM=DCOS(0MEGA)
50. C SENNE=DSIN(N)
51. C CBSNE=DCOS(N)
52. C AI=ARCS(DABS(.99597036*CBSBM+.08968308*SENBM*CBSNE))
53. C SENI=DSIN(AI)
54. C SENNU=.08968308D0*SENNE/SENI
55. C FNU=ARSIN(SENNU)
56. C SENAL=SENBM*SENNE/SENI
57. C CBSAL=CBSNE*DCOS(FNU)+SENNE*SENNU*CBSBM
58. C CASBL=1.D0+CBSAL
59. C L=S1+N+2.D0*DATAN2(SENAL,CASBL)

```

```

60. DE1=.03350208D0+.000084D0*CENT
61. L1=H+DE1*DSIN(H*P1)
62. GH=57.295780*M
63. GLONG=-57.295780*RLONG
64. TH=FLOAT(IH)+FLOAT(IM)/60.
65. CHI1=.01745329D0*(TH*15.D0-180.D0-GLONG*GH)
66. CHI=CHI1*FNU
67. CSZ=DSIN(RLAT)*SENI*DSIN(L)+DCOS(RLAT)*((DCOS(AI/2.D0)**2*DCOS(L-C
68. HI)+DSIN(AI/2.D0)**2*DCOS(L+CHI)))
69. CST=DSIN(RLAT)*SENM*DSIN(L1)+DCOS(RLAT)*(.9587251D0*DCOS(L1-CHI1
70. 1)+.0412749D0*DCOS(L1+CHI1))
71. DMUN=2.6014433D-11+1.432503D-12*DCOS(SP)+7.8644D-14*DCOS(DSP)+2.00
72. 1919D-13*DCOS(SHP)+1.46007D-13*DCOS(CSH)
73. DSUN=6.6889632D-14+1.118955D-15*DCOS(H*P1)
74. RH0=6.378388D+8*(1.D0+.003367D0*DSIN(RLAT)**2)
75. PC1=5.886D+21*DMUN
76. PC1=PC1*RH0*DMUN
77. FC1=PC1*DMUN
78. PC2=8.829D+21*DMUN
79. PC2=PC2*DMUN
80. PC2=PC2*DMUN*RH0
81. PC2=PC2*DMUN*RH0
82. PC3=1.5952D+29*DSUN
83. PC3=PC3*RH0*DSUN
84. PC3=PC3*DSUN
85. CLS=PC1*(3.D0*CSZ**2-1.D0)+PC2*(5.D0*CSZ**3-3.D0*CSZ)+PC3*(3.D0*CS
86. 1T**2-1.D0)
87. WBNK= 0.037D0*(3.0D0*DSIN(RLAT)**2-1.0)
88. RETURN
89. END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
AI	C	SCALR	0004E V	2	ARCBS	D	SPR0G	0000A S	1	ARSIN	D	SPR0G	0000A S	1
CASBL	D	SCALR	00056 V	2	CENT	D	SCALR	00012 DN	2	CHI	D	SCALR	0003A V	2
CHI1	D	SCALR	0003C V	2	CLS	R	SCALR	0001E SP	DUMMY	COSAL	D	SCALR	00032 V	2
CBSNE	D	SCALR	0002C V	2	CBS0M	D	SCALR	0002A SENNE	2	CST	D	SCALR	00040 V	2
CSZ	D	SCALR	0003E V	2	DABS	C	SPR0G	00036 L	2	DATAN2	D	SPR0G	00040 V	2
CBIS	D	UNUSED	0000C V	2	DCBS	C	SPR0G	00042 DMUN	2	DE1	D	SCALR	00034 V	2
DM	D	SCALR	0000F V	1	DIBIS	D	SCALR	0004E AI	2	DID	D	SCALR	00004 V	2
DMUN	D	SCALR	00042 V	2	DIY	D	SCALR	00050 V	2	DROD	D	SPR0G	00004 V	2
DP1	D	SCALR	00014 V	2	DN	D	SCALR	00054 V	2	DP	D	SCALR	00010 V	2
DSIN	D	SPR0G	00018 V	2	DS	D	SCALR	00058 V	2	DSH	D	SCALR	00022 V	2
DTD	D	SCALR	0006B V	DUMMY	DSP	D	SCALR	00064 V	2	DSUN	D	SCALR	0004C V	2
FNU	D	SCALR	00052 V	2	DYP	D	SCALR	00068 V	1	FLGAT	R	SPR0G	00004 V	1
H	D	SCALR	0005E V	DUMMY	GH	R	SCALR	0006A V	1	GLONG	I	SCALR	0005E V	1
ID	I	SCALR	00066 V	DUMMY	HOK	R	SCALR	00066 V	1	IBIS	I	SCALR	0005E V	1
INT	I	SPR0G	00066 V	DUMMY	IV	I	SCALR	00067 V	DUMMY	IM	I	SCALR	00068 V	DUMMY
L1	D	SCALR	00038 V	2	IV	I	SCALR	00065 V	DUMMY	L	D	SCALR	00036 V	2
P	D	SCALR	00018 V	2	N	D	SCALR	00054 V	2	MEGA	D	SCALR	0001A V	2
PC3	D	SCALR	0004A V	2	PC1	D	SCALR	00046 V	2	PC2	D	SCALR	00048 V	2
RLAT	D	SCALR	00063 V	DUMMY	P1	D	SCALR	00016 V	2	RH0	D	SCALR	00044 V	2
SENAL	D	SCALR	00030 V	2	RLONG	D	SCALR	00064 V	2	S	D	SCALR	0000A V	2
SENNU	D	SCALR	0002E V	2	SEN1	D	SCALR	00050 V	2	SENNE	D	SCALR	0002A V	2
SP	D	SCALR	0001C V	2	SEN0M	D	SCALR	00026 V	2	SHP	D	SCALR	00020 V	2
TIDAL	R	SCALR	0000C V	1	S1	D	SCALR	00024 V	2	TH	R	SCALR	00062 V	1
X1	R	SCALR	0005D V	1	TICAL	D	SPR0G	00000 P	2	X	R	SCALR	0005C V	1

## LOCAL VARIABLES (99 WORDS):

0000C TICAL	00002 DLY	00004 DID	00006 CENT	00008 DS	0000A S
0000C DH	0000E W	00010 DP	00012 DN	00014 DP1	00016 P1
00018 P	0001A MEGA	0001C SP	0001E SP	00020 SHP	00022 DSH
00024 S1	00026 SEN0M	00028 CBS0M	0003A SENNE	00032 COSNE	00034 SENNU
0003C SENAL	0003E CBSAL	00040 DE1	00042 DMUN	00044 L1	00046 CHI
00048 PC1	00050 CSZ	00054 CST	00058 AI	00062 RH0	00064 PC1
00054 N	00056 CASOL	00060 DYM	00064 DIBIS	00066 SENI	00068 FNU
0005E IBIS	0005F CIH	0006C GH	00066 GLENG	00068 X	00070 X1

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

0000C TICAL

## INTRINSIC SUBPROGRAMS USED:

CABS	DATAN2	DCBS	DM0C	DSIN	FL0AT	INT
ARCBS	ARSIN	9CABS	9DSIN	9CT0R	9IT0D	9IT0R
SRT0I	9SETUPN					

## EXTERNAL SUBPROGRAMS REQUIRED:

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	450	001C2
CONSTANTS:	116	00074
LOCAL VARIABLES:	99	00063
TEMPS:	22	00016
TOTAL PROGRAM:	687	002AF

```

1.      SUBROUTINE VETBY
2.      A( IDATA,IE0D,IIN,IIBUT,ITAPE,NUMPL,DATA,RLAT,RLONG,K0GHM,IAGAP,LCN
3.      BT,RACEG,DEGRA,KDEG2,IDEG2,FDEG2,RDEG2,RTOP,ITOP,RBOT,IBOT,LEFT,
4.      C ILEFT,RRIGT,IRIGT,SLTK,SLGK,SINCH,SMP,FBOT,FTOP,FLEFT,FRIGT,NDEG,
5.      D SLAT,SLONG,B0TMP,XX,YY,INIT,X0LD,Y0LD)
6.      C
7.      C      SUBROUTINE VETBY, TO SET CHART BOUNDARIES BY
8.      C      OPERATOR ENTRY ON CONSOLE TTY
9.      C
10.     C      FOR NON INTEGER CHART BOUNDARIES
11.     C
12.     C      CALLS SUBROUTINE ARLIM
13.     C
14.     C
15.     C      CALL ARLIM(IIN,IIBUT,RTOP,RBOT,RLEFT,RRIGT)
16.     C      SMP=SINCH/60.0
17.     C      RDEG2=(RRIGT-RLEFT)
18.     C      FDEG2=RDEG2*RADEG
19.     C      RETURN
20.     C      END

```

NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS	NAME	TYPE	CLASS	HEX L6C	DEC WORDS
ARLIM		SPRGG	EXTERNAL		B8TMP		UNUSED*00C26	V	DUMMY	DATA		UNUSED*00007	V	DUMMY
DEGRA		UNUSED*0000E	V	DUMMY	FB8T		UNUSED*0001F	V	DUMMY	FDEG2	R	SCALR*00011	V	DUMMY
FLEFT		UNUSED*00021	V	DUMMY	FRIGT		UNUSED*00C22	V	DUMMY	FT8P		UNUSED*00020	V	DUMMY
IAGAP		UNUSED*0000B	V	DUMMY	IB8T		UNUSED*00016	V	DUMMY	IDATA		UNUSED*00001	V	DUMMY
IDEG2		UNUSED*00010	V	DUMMY	IE8C		UNUSED*00CC2	V	DUMMY	IIN	I	SCALR*00003	V	DUMMY
IIBLT	I	SCALR*00004	V	DUMMY	ILEFT		UNUSED*00C18	V	DUMMY	INIT		UNUSED*00029	V	DUMMY
IRIGT		UNUSED*0001A	V	DUMMY	ITAPE		UNUSED*00CC5	V	DUMMY	IT8P		UNUSED*00014	V	DUMMY
KDEG2		UNUSED*0000F	V	DUMMY	K8GHM		UNUSED*0000A	V	DUMMY	LCNT		UNUSED*0000C	V	DUMMY
NDEG2		UNUSED*00023	V	DUMMY	NUMPL		UNUSED*00CC6	V	DUMMY	RADEG	R	SCALR*0000D	V	DUMMY
RB8T	R	SCALR*00015	V	DUMMY	RDEG2		UNUSED*00012	V	DUMMY	RLAT		UNUSED*00008	V	DUMMY
RLEFT	R	SCALR*00017	V	DUMMY	RLONG		UNUSED*00009	V	DUMMY	RRIGT	R	SCALR*00019	V	DUMMY
RT8P	R	SCALR*00013	V	DUMMY	SINCH		UNUSED*0001D	V	DUMMY	SLAT		UNUSED*00024	V	DUMMY
SLGK		UNUSED*0001C	V	DUMMY	SLONG		UNUSED*00C25	V	DUMMY	SLTK		UNUSED*0001B	V	DUMMY
SMP	R	SCALR*0001E	V	DUMMY	VETBY		SPRGG 00000	P	DUMMY	VETBY	R	SCALR 00000	V	DUMMY
X8LD		UNUSED*0002A	V	DUMMY	XX		UNUSED*00C27	V	DUMMY	Y8LD		UNUSED*0002B	V	DUMMY
YY		UNUSED*00028	V	DUMMY										

## LOCAL VARIABLES (1 WORD):

00000 VETBY

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

00000 VETBY

## EXTERNAL SUBPROGRAMS REQUIRED:

ARLIM 9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	64	00040
CONSTANTS:	1	00001
LOCAL VARIABLES:	1	00001
TEMPS:	44	0002C
TOTAL PROGRAM:	110	0006E

```

1.      SUBROUTINE WEIG2(XP0L,ZP0L,NVERT,X,NPTS,SUM,RHO,TEST,DSU)
2.      C      THIS IS VERSIONS 2 WHICH ALSO DOES WEIGHTTEST
3.      C      THIS SUBROUTINE IS TO BE USED WITH TALPL0T. IT COMPUTES
4.      C      THE DENSITY CONTRIBUTION OF A POLYGON OF DENSITY RHO
5.      C      AND ADDS THE CONTRIBUTION TO THE SUM.
6.      C      XP0L, ZP0L ARE THE COORDINATES OF THE VERTICES OF THE POLYGON
7.      C      NVERT IS THE # OF VERTICES IN A POLYGON
8.      C      X IS THE COORDINATE AT WHICH WE WISH THE SM CALCULATED.
9.      C      NPTS IS THE # OF POINTS AT WHICH WE WISH THE SUM CALCULATED
10.     C      SUM IS THE ACCUMULATED DENSITY CONTRIBUTION
11.     C      A RESTRICTION IS THAT THE FIRST THREE (3) POINTS OF A POLYNOMIAL
12.     C      MAY NOT HAVE THE SAME X COORDINATE. • THE FIRST TWO(2) MAY
13.     C      BE THE SAME, AND AFTER THE FIRST VERTEX ANY NUMBER MAY
14.     C      THE DIMENSION OF XP0L,ZP0L, MUST BE 3 GREATER IN THE MAIN
15.     C      PROGRAM THEN THE ACTUAL NO OF VERTICES (NVERT)
16.     C      DIMENSION XP0L(1),ZP0L(1),X(1),Z(1),WT(1),SUM(1),NFLAG(10),
17.     C      1DIS(10),S0RT(10),KFLAG(10),TEST(1),DSL(1)
18.     C      ZP0L(NVERT+1)=ZP0L(2)
19.     C      ZP0L(NVERT+2)=ZP0L(3)
20.     C      ZP0L(NVERT+3)=ZP0L(4)
21.     C      XP0L(NVERT+1)=XP0L(2)
22.     C      XP0L(NVERT+2)=XP0L(3)
23.     C      XP0L(NVERT+3)=XP0L(4)
24.     C      DO 300 I=1,NPTS
25.     C      SU=0.
26.     C      INTER=1
27.     C      II0UT=108
28.     C      NDUM=NVERT+2
29.     C      JJ=3
30.     C      XX=X(I)
31.     C      DO 9 IG=1,10
32.     C      DIS(IG)=0.
33.     5      NFLAG(IG)=-1.
34.     C      IF (XX-XP0L(3)) 11,15,80
35.     15      JJ=2
36.     C      NDUM=NVERT+1
37.     C      IF (XX-XP0L(2)) 11,14,80
38.     14      JJ=1
39.     C      NDUM=NVERT
40.     C      IF (XX-XP0L(1)) 11,17,80
41.     17      CONTINUE
42.     C      WRITE(II0UT,18)
43.     18      FORMAT(' ***** FIRST 3 VERTICES HAVE •EQ. X COORD. ' )
44.     10      CONTINUE
45.     11      CONTINUE
46.     12      JJ=JJ+1
47.     C      IF (JJ.GT.NDUM ) GO TO 100
48.     C      IF ( XX-XP0L(JJ) ) 11,20,21
49.     20      JAC=JJ
50.     22      IF (XX.NE.XP0L(JJ+1) ) GO TO 24
51.     C      JJ=JJ+1
52.     C      GO TO 22
53.     21      DIS(INTER)=((XP0L(JJ)-XX)*ZP0L(JJ-1)+(XX-XP0L(JJ-1))*ZP0L(JJ))
54.     C      C/((XP0L(JJ)-XP0L(JJ-1)))
55.     C      INTER=INTER+1
56.     C      GO TO 80
57.     C      THIS SECTION HANDLES INTERSECTION WITH A VERTICAL
58.     C      LINE OR INTERSECTION THRU ONE OF THE VERTICES OF THE POLYGON
59.     24      IF (XP0L(JJ+1).GT.XX) GO TO 26

```

```

60.      IF (JU.GE.NDUM ) GO TO 100
61.      DIS(INTER)=(ZPBL(JU)+ZPBL(JAC))/2.
62.      INTER=INTER+1
63.      GO TO 80
64.  26   IF (JAC.EG.JU) GO TO 11
65.      DIS(INTER)=ZPBL(JAC)
66.      NFLAG(INTER)=INTER
67.      INTER=INTER+1
68.      DIS(INTER)      =ZPBL(JU)
69.      NFLAG(INTER)=INTER-1
70.      INTER=INTER+1
71.      GO TO 11
72.  80   CONTINUE
73.      JU=JU+1
74.      IF (JU.GT.NDUM ) GO TO 100
75.      IF (XPBL(JU)-XX) 80,90,91
76.  90   JAC=JU
77.  92   IF (XX.NE.XPBL(JU+1) ) GO TO 94
78.      JU=JU+1
79.      GO TO 92
80.  91   DIS(INTER)=((XX-XPBL(JU))*ZPBL(JU-1)+(XPBL(JU-1)-XX)*ZPBL(JU))
81.      1/(XPBL(JU-1)-XPBL(JU))
82.      INTER =INTER+1
83.      GO TO 11
84.  94   IF ( XPBL(JU+1).LT.XX) GO TO 96
85.      DIS(INTER)=(ZPBL(JU)+ZPBL(JAC))/2.
86.      INTER=INTER+1
87.      GO TO 11
88.  96   IF(JAC.EG.JU) GO TO 80
89.      DIS(INTER)=ZPBL(JAC)
90.      NFLAG(INTER)=INTER
91.      INTER=INTER+1
92.      DIS(INTER)=ZPBL(JU)
93.      NFLAG(INTER)=INTER-1
94.      INTER=INTER+1
95.      GO TO 80
96.  100  CONTINUE
97.  C    WRITE (108,517)
98.  517  FORMAT('DIS' (NFLAG'))
99.  C
100. C
101. C
102. C    WE HAVE NOW LOCATED ALL THE INTERSECTIONS WHICH RUN DOWN THE
103. C    BODY OF A POLYGON AND NEVER CROSSES IN OR OUT
104. C    THE INTERSECTION WILL NOW BE SORTED FROM SMALLEST TO LARGEST
105. C    INTER=INTER-1
106. C    THIS CHANGES INTER SO THAT IT NOW = THE # OF INTERSECTIONS
107. C    IF(INTER.EG.C) GO TO 300
108. C    IF THERE ARE NO INTERSECTIONS WE BYPASS THE COMPUTATION
109. C    OF THE SDISTANCE
110. C    SORT FROM SMALLEST TO LARGEST
111. C    DO 112 IU=1,INTER
112. C    JUL=1
113. C    KFLAG(IL)=NFLAG(1)
114. C    SORT(IL)=DIS(1)
115. C    DO 110 JU=2,INTER
116. C    IF(SORT(IU).LE.DIS(JU)) GO TO 110
117. C    SORT(IL)=DIS(JU)
118. C    KFLAG(IL)=NFLAG(JU)
119. C    JUL=JU

```

```

120.      110  CONTINUE
121.      DIS(JUL)=1.E70
122.      112  CONTINUE
123.      SUBT=0.
124.      IF(SORT(1)) 2201,2202,2202
125.      2201  CONTINUE
126.      DSOR=0.
127.      IF(SORT(2).LT.0) DSOR=SORT(2)
128.      SUBT=(SORT(1)-DSOR)
129.      2202  CONTINUE
130.      C      WRITE (108,52) INTER
131.      C      WRITE (108,518),(SORT(IX),KFLAG(IX),IX=1,4)
132.      52     FORMAT (1X,I3)
133.      518    FORMAT(1X,F6.3,1X,I3)
134.      C      THE NB ARE ALL SORTED NOW
135.      C      WE ARE NOW GOING TO COMPLETE THE SI DISTANCE
136.      C
137.      MDID=0
138.      201   IF(INTER=MDID) 999,999,202
139.      202   MDID=MDID+1
140.      IF(KFLAG(MDID)) 203,203,221
141.      203   SU      =SU      +SORT(MDID+1)-SORT(MDID)
142.      IF (      KFLAG(MDID+1))204,204,245
143.      204   MDID=MDID+1
144.      GO TO 201
145.      C      THIS HAS NOW HANDLED THE NORMAN SECTION
146.      221   IF (      KFLAG(MDID).NE.KFLAG(MDID+1)) GO TO 224
147.      SU      =SU      +(SORT(MDID+1)-SORT(MDID))/2.
148.      MDID=MDID+1
149.      GO TO 201
150.      224   SU      =SU      +(SORT(MDID+3)+SORT(MDID+2)-SORT(MDID+1)-SORT(MDID))
151.      C/2.
152.      MDID=MDID+3
153.      GO TO 201
154.      245   IF(      KFLAG(MDID+1).NE.KFLAG(MDID+2)) GO TO 248
155.      SU      =SU      +(SORT(MDID+2)-SORT(MDID+1)) /2.
156.      MDID=MDID+2
157.      GO TO 203
158.      248   SU      =SU      +(SORT(MDID+2)-SORT(MDID+1)+SORT(MDID+4)-SORT(MDID+3)
159.      8)/2.
160.      MDID=MDID+4
161.      GO TO 203
162.      999   SUM(I)=SUM(I)+RH0*SU      *100.
163.      TEST(I)=TEST(I)+(SU+SUBT)*267.
164.      CSL(I)=SU*RH0*100.
165.      300   CONTINUE
166.      RETURN
167.      END

```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
DIS	I	SCALR	0000D	10	DSR	I	SCALR	00038	V	DSU	I	SCALR	00042	V	INTER	I	SCALR	0002D	V
I	I	SCALR	0002B	V	IUBT	I	SCALR	0002E	V	JAC	I	SCALR	00033	V	JU	I	SCALR	00036	V
IC	I	SCALR	00032	V	U	I	SCALR	00034	V	NDUM	I	SCALR	0002F	V	NVERT	I	SCALR	0003C	V
JU	I	SCALR	00030	V	MDID	I	SCALR	00039	V	SU	R	SCALR	0002C	V	TEST	R	SCALR	00041	V
KFLAG	I	ARRAY	00021	V	NPTS	I	SCALR	0003E	V	WT	R	ARRAY	00002	V	XX	R	SCALR	00031	V
NFLAG	I	ARRAY	00003	V	SBRT	R	ARRAY	00017	V	XPBL	R	SCALR	0003A	V					
R-B	R	SCALR	00040	V	WEIG2	R	SCALR	0003F	V	ZPBL	R	ARRAY	0003B	V					
SBRT	R	SCALR	00037	V	SUP	R	ARRAY	0003A	V										
WEIG2	R	SCALR	00000	F	XPBL	R	SCALR	0003A	V										
X	R	ARRAY	0003D	V	ZPBL	R	ARRAY	0003B	V										
Z	R	ARRAY	00001	V															

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
9	CC044	11	CC072	12	CC072	14	00057	15	0004D
17	CC06C	20	CC07B	21	CC084	22	0007D	24	00098
26	CC0A9	80	CC0BE	90	CC0C7	91	000D0	92	000C9
34	CC0E6	100	CC109	110	CC124	112	CC12E	201	00149
202	CC14C	204	CC150	221	CC15A	224	00165	245	0016F
248	CC17A	517	CC109	518	CC142	999	00194	2201	00136
2202	CC13F								

## LOCAL VARIABLES (58 WORDS):

0000C	WEIG2	00001	Z
00021	KFLAG	0002B	I
00030	JU	00031	XX
00036	JU	00037	SBRT

BLANK COMMON (0 WORDS)

ENTRY POINTS:

00000 WEIG2

## EXTERNAL SUBPROGRAMS REQUIRED:

F:102	F:104	F:106	F:108	9BCDWRT	SENDI8L	9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC	HEX
WORDS	WORDS
409	00199
4	00004
58	0003A
12	0000C
483	001E3

GENERATED CODE:

CONSTANTS:

LOCAL VARIABLES:

TEMPS:

TOTAL PROGRAM:

```

1. C SUBROUTINE YBLKI BLOCKED TAPE INPLT SEIS FORMAT
2. C MODIFIED MAY 13, 1971 BY REMOVING RECORD SKIP AFTER BAD READ (FOR
3. C NEW MONITOR)
4. C MODIFIED TO YBLKI FROM GBLKI MARCH 25, 1971 BY FOLINSBEE
5. C PROGRAMMER J WEBSTER GBLI 1
6. C FOR C BOWIN GBLI 2
7. C NOV 1970 GBLI 3
8. C GBLI 4
9. C*****GBLI 5
10. C SUBROUTINE HAS TWO ENTRY POINTS, YSETI, AND YBLKI
11. C YSETI IS THE INITIALIZING ENTRY, YBLKI IS THE NORMAL ENTRY GBLI
12. C SUBROUTINE READS BLOCKED RECORDS FROM A MAG TAPE, GBLI 8
13. C AND RETURNS DATA FROM ONE LOGICAL RECORD, GBLI 9
14. C CONVERTED ACCORDING TO A SPECIFIED FORMAT GBLI 10
15. C*****GBLI 11
16. C SUBROUTINE YBLKI(
17. C 1 ISR1,ISR2,KDA,KMB,KYR,KHM,ISEC,ILAT,KSX,ILBN,KWE,IDEPT,IAMAG,IMB,
18. C 2 ISOS,INTS,IDIAS,ITSU,ISEICH,IVBLK,INONT,IWG,IFEG,IMS,IASP,IZH,
19. C 3 ICE,IMG,IAUTH,IGHY,NPF,IQG,ILM,IS1,IS2
20. C 4 )
21. C GBLI 15
22. C ALL YBLKI ARGUMENTS ARE VARIABLE NAMES FOR DATA
23. C TO BE READ FROM 1 LOGICAL RECORD GBLI 17
24. C GO TO 100 GBLI 18
25. C GBLI 19
26. C*****GBLI 20
27. C ENTRY YSETI(ITAPE,IFMT,INDIC,IBLFC,Irlen,IBUF)
28. C DIMENSION IBUF(1)
29. C GBLI 22
30. C THIS IS THE INITIALIZING ENTRY GBLI 23
31. C ITAPE IS LOGICAL UNIT NUMBER FOR INPLT GBLI 24
32. C IFMT IS STATEMENT NO. OF FORMAT GBLI 25
33. C INDIC IS INDICATOR OF INPLT STATUS GBLI 26
34. C 1=READ OKAY GBLI 27
35. C 2=END OF FILE GBLI 28
36. C 4=PARITY ERROR GBLI 29
37. C 5=FORMAT ERROR GBLI 30
38. C 6=BOTH 4 AND 5 TYPE ERRORS FOUND GBLI 31
39. C GBLI 32
40. C IBLFC IS BLOCKING FACTOR (NO OF LOGICAL RECORDS PER BLOCK) GBLI 33
41. C Irlen IS LOGICAL RECORD LENGTH (MUST BE MULTIPLE OF 4) GBLI 34
42. C IBUF IS INPLT BUFFER GBLI 35
43. C IF IBLFC AND Irlen ARE CHANGED, THE SIZE OF ARRAY IBUF GBLI 36
44. C MUST ALSO BE CHANGED TO IBLFC*Irlen/4 GBLI 37
45. C IWORD=Irlen/4 GBLI 41
46. C IBLSZ=IBLFC*IWORD GBLI 42
47. C ICNT=0 GBLI 43
48. C RETURN GBLI 44
49. C END OF INITIALIZING PART OF SUBROUTINE GBLI 45
50. C GBLI 46
51. C*****GBLI 47
52. C SET UP RUN-TIME ABORT FOR FORMAT ERRORS GBLI 48
53. C 100 CALL ABORTSET(2205,6) GBLI 49
54. C IF(INDIC.EQ.6) INDIC=4
55. C IF(INDIC.EQ.5) INDIC=1
56. C ITOTR=IBLFC GBLI 50
57. C GBLI 51
58. C TEST WHETHER BLOCK IS TO BE READ FROM TAPE GBLI 52
59. C IF(ICNT.NE.0) GO TO 125 GBLI 53

```

60.	C		GBLI 54
61.	C	YES, READ PHYSICAL RECORD	GBLI 55
62.		CALL BUFFERIN(ITAPE, 0, IBUF, IBLSZ, IND, NW)	GBLI 56
63.	110	GO TO (110, 120, 200, 210), IND	GBLI 57
64.	120	INDIC=1	GBLI 58
65.	125	IF(NW.NE. IBSZ) ITOTR=NW/IRLEN	
66.		*WRITE(108, 501) NW	
67.	501	FORMAT('YBLKI: ABNORMAL REC LENGTH, NW=', I10 )	
68.	126	CONTINUE	
69.		ICNT=ICNT+1	GBLI 60
70.	C		GBLI 61
71.	C	DECODE LOGICAL RECORD ACCORDING TO FORMAT STATEMENT	GBLI 62
72.		DECODE(IRLEN, IFMT, IBUF( ))	GBLI 63
73.		1 ISR1, ISR2, KDA, KMB, KYR, KHY, ISEC, ILAT, KSN, ILON, KWE, IDEPT, JAMAG, IMB,	
74.		2 ISOS, INTS, IDIAS, ITSU, ISEICH, IVOLC, INONT, IWG, IFEG, IMS, IASP, IZH,	
75.		3 ICE, IMG, IALTH, IQHY, NPP, IQG, ILM, IS1, IS2	
76.	C		GBLI 67
77.	C	KEEP TRACK OF NO. OF LOGICAL RECORDS	GBLI 68
78.	130	CONTINUE	
79.		ICNT=ICNT+1	GBLI 69
80.		IF(ICNT.GE.ITOTR) ICNT=C	GBLI 70
81.		CALL ABORTSET(0)	
82.		RETURN	GBLI 72
83.	C		GBLI 73
84.	C	CONTROL TRANSFERS HERE FOR EOF	GBLI 74
85.	200	INDIC=2	GBLI 75
86.		GO TO 130	GBLI 76
87.	C		GBLI 77
88.	C	CONTROL COMES HERE FOR READ ERROR	GBLI 78
89.	210	INDIC=4	GBLI 79
90.		OUTPUT 'YBLKI READ ERROR'	
91.	C		
92.	C	RESET NW (COMPENSATING FOR BUFFER ERROR)	
93.		NW=IBLSZ	
94.	C		
95.		GO TO 126	
96.	C		GBLI 81
97.	C	CONTROL COMES HERE FOR FORMAT ERROR	GBLI 82
98.	220	CONTINUE	
99.		IF(INDIC.EQ.4) INDIC=6/GO TO 130	
100.		INDIC=5	GBLI 84
101.		GO TO 130	GBLI 85
102.		END	GBLI 86

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
ABORTSET	I	SPR0G	00021	DUMMY	BUFFERIN	I	SPR0G	00025	DUMMY	IAMAG	I	SCALR	00015	DUMMY
IASP	I	SCALR	00008	1	IATH	I	SCALR	00025	DUMMY	IBLFC	I	SCALR	0002F	DUMMY
IBLSZ	I	SCALR	00023	DUMMY	IBLSZ	I	SCALR	000C2	1	IBUF	I	ARRAY	00031	DUMMY
ICE	I	SCALR	00019	DUMMY	ICAT	I	SCALR	000C3	1	IDPT	I	SCALR	00014	DUMMY
IDIAS	I	SCALR	0001C	DUMMY	IFEG	I	SCALR	000C1F	DUMMY	IFMT	I	SCALR	0002D	DUMMY
ILAT	I	SCALR	00016	DUMMY	ILM	I	SCALR	000C29	DUMMY	ILBN	I	SCALR	00012	DUMMY
IME	I	SCALR	00005	DUMMY	IMG	I	SCALR	000C24	DUMMY	IMS	I	SCALR	00020	DUMMY
IND	I	SCALR	00018	DUMMY	INDIC	I	SCALR	000C2E	DUMMY	INNT	I	SCALR	0001D	DUMMY
INTS	I	SCALR	00030	DUMMY	INSEC	I	SCALR	000CF	DUMMY	IGMY	I	SCALR	00026	DUMMY
IRLEN	I	SCALR	00017	DUMMY	IS1	I	SCALR	00009	DUMMY	ISEICH	I	SCALR	0001B	DUMMY
IS1	I	SCALR	0002A	DUMMY	IS2	I	SCALR	0002B	DUMMY	ISR2	I	SCALR	0000A	DUMMY
IT0TR	I	SCALR	00004	1	ITSU	I	SCALR	0001A	DUMMY	ITAPE	I	SCALR	0002C	DUMMY
ING	I	SCALR	0001E	DUMMY	IWRD	I	SCALR	0000B	DUMMY	IV0LC	I	SCALR	0001C	DUMMY
KM8	I	SCALR	0000C	DUMMY	KDA	I	SCALR	00011	DUMMY	IZH	I	SCALR	00022	DUMMY
KYR	I	SCALR	0000D	DUMMY	KSN	I	SCALR	00011	DUMMY	KHM	I	SCALR	0000E	DUMMY
YBLKI	R	SCALR	00000	1	NPP	I	SCALR	00027	DUMMY	KWE	I	SCALR	00013	DUMMY
					YBLKI	I	SPR0G	00000	P	NW	I	SCALR	00006	DUMMY
										YSETI	I	SPR0G	00028	P

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
100	0003B	120	00061	125	00063	126	00078
200	0008C	220	000C2	501	0006D	130	000A6

## LOCAL VARIABLES (9 WORDS):

0000C YBLKI	000C1 IWRD	00004 IT0TR	00005 IND
00006 NW	00007 W		

## BLANK COMMON (0 WORDS)

## ENTRY POINTS:

0000C YBLKI	00028 YSETI
-------------	-------------

## EXTERNAL SUBPROGRAMS REQUIRED:

ABORTSET	BUFFERIN	F:108	9AS0RM	9DECODE	SENDI0L	918DATA	9PRINT
9SETUPN							

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
GENERATED CODE:	204	C0CCC
CONSTANTS:	3	C0C03
LOCAL VARIABLES:	9	C0009
TEMPS:	43	C0C2B
TOTAL PROGRAM:	259	CC103

```

1.  C      SUBROUTINE YBLK0 - BLOCKED TAPE OUTPUT
2.  C      VERSION OF APRIL 11, 72 TO OUTPUT LAT AND LONG KEY AT END
3.  C      OF RECORD
4.  C      MOD OF AUG 16, 71 TO OUTPUT SHORT RECORD WHEN YEND0 CALLED
5.  C      MODIFIED FROM GBLK0 ON MAR 25 71 BY AFOLINSBEE TO OUTPUT SEIS DATA
6.  C      PROGRAMMER J WEBSTER
7.  C      FOR C BSWIN
8.  C      NOV 1970
9.  C
10. C*****
11. C      SUBROUTINE HAS 3 ENTRY POINTS: YSET0, YBLK0, YEND0
12. C      YSET0 IS THE INITIALIZING ENTRY
13. C      YBLK0 IS THE NORMAL ENTRY
14. C      YEND0 IS THE TERMINATING ENTRY
15. C*****
16. C      SUBROUTINE YBLK0(
17. C      1 ISR1,ISR2,KDA,KM0,KYR,KHY,ISEC,ILAT,KSX,ILON,KWE,IDEPT,IAMAG,IMB,
18. C      2 ISOS,INTS,IDIAS,ITSU,ISEICH,IVOLC,INONT,ING,IFEG,IMS,IASP,IZH,
19. C      3 ICE,IMG,IALTH,IGHY,NPF,IOG,ILM,IS1,IS2
20. C      4 )
21. C      ALL YBLK0 ARGUMENTS ARE VARIABLE NAMES FOR DATA
22. C      TO BE WRITTEN IN 1 LOGICAL RECORD
23. C      DATA JNN/1HN/
24. C      DATA JEE/1HE/
25. C      GO TO 300
26. C*****
27. C      ENTRY YSET0(JTAPE,JFMT,NDPS,JFULL,JBLFC,JRLEN,MAXBL,JBUF)
28. C      DIMENSION JBUF(1)
29. C
30. C      THIS IS THE INITIALIZING ENTRY
31. C      JTAPE IS THE LOGICAL UNIT NUMBER FOR OUTPUT
32. C      JFMT IS STATEMENT NO. OF FORMAT
33. C      NDPS IS THE NO. OF DATA POINTS CURRENTLY WRITTEN ON A TAPE
34. C      JFULL IS STATEMENT NO. TO WHICH CONTROL IS TRANSFERRED
35. C      WHEN OUTPUT TAPE IS FULL
36. C
37. C      JBLFC IS BLOCKING FACTOR (NO. OF LOGICAL RECORDS PER BLOCK)
38. C      JRLEN IS LOGICAL RECORD LENGTH (MUST BE MULTIPLE OF 4)
39. C      JBUF IS OUTPUT BUFFER
40. C      IF JBLFC AND JRLEN ARE CHANGED, THE SIZE OF ARRAY JBUF
41. C      MUST BE CHANGED TO JBLFC*JRLEN/4
42. C
43. C      IIOUT=108
44. C
45. C      MAXBL IS MAXIMUM NO. OF BLOCKS PER TAPE
46. C
47. C      JBLK COUNTS BLOCKS
48. C      JBLK=0
49. C
50. C      ZERO OUT DATA POINT COUNTER
51. C      NDPS=0
52. C      JCNT COUNTS LOGICAL RECORDS WITHIN A BLOCK
53. C      JCNT=0
54. C      JWORD=JRLEN/4
55. C      JBSZ=JBLFC*JWORD
56. C      RETURN
57. C
58. C      END OF INITIALIZATION
59. C*****

```

```

GBL0 1
GBL0 2
GBL0 3
GBL0 4
GBL0 5
GBL0 10
GBL0 15
GBL0 16
GBL0 17
GBL0 19
GBL0 20
GBL0 21
GBL0 22
GBL0 23
GBL0 24
GBL0 25
GBL0 26
GBL0 27
GBL0 28
GBL0 29
GBL0 30
GBL0 31
GBL0 32
GBL0 36
GBL0 37
GBL0 38
GBL0 40
GBL0 41
GBL0 42
GBL0 43
GBL0 44
GBL0 45
GBL0 46
GBL0 47
GBL0 48
GBL0 49
GBL0 50
GBL0 51
GBL0 52
GBL0 53

```

60.	C	THIS IS THE TERMINATING ENTRY POINT	GBL0 54
61.		ENTRY YEND0	
62.	C		GBL0 57
63.		IF(JCNT.EG.0) G0 T0 400	
64.	C	SETTING UP T0 OUTPUT SHORT RECORD	
65.		JBLSZ=JWORD*JCNT	
66.		G0 T0 310	GBL0 62
67.	C	*****	GBL0 63
68.	C		GBL0 64
69.	C	CONVERT LOGICAL RECORD T0 BCD	GBL0 65
70.	300	I=JCNT*JWORD+1	GBL0 66
71.		TKEY=ILAT/1000	
72.		GKEY=IL0N/1000	
73.		IF(KSN.NE.JNN) TKEY=-TKEY	
74.		IF(KWE.NE.JEE) GKEY=-GKEY	
75.		LTKEY=TKEY+90.	
76.		LGKEY=GKEY+180.	
77.		ENCODE(JRLEN,JFMT,JBUF(I))	GBL0 67
78.		1 ISR1,ISR2,KDA,KMB,KYR,KHP,ISEC,ILAT,KSN,IL0N,KWE,IDEPT,IAMAG,IMB,	
79.		2 IS0S,INTS,IDIAS,ITSU,ISEICH,IV0LC,IN0NT,IWG,IFEG,IMS,IASP,IZH,	
80.		3 ICE,IMG,IAUTH,IGHY,NPF,I0G,ILM,ISI,IS2,LTKEY,LGKEY	
81.		JCNT=JCNT+1	GBL0 71
82.	C		GBL0 72
83.	C	TEST IF READY T0 WRITE BLOCK T0 TAPE	GBL0 73
84.		IF(JCNT.LT.JBLFC) G0 T0 400	GBL0 74
85.	C		GBL0 75
86.	C	WRITE BLOCK 0NT0 TAPE	GBL0 76
87.	310	CALL BUFFER 0UT (JTAPE,C,JBUF,JBLSZ,IND)	GBL0 77
88.		JBLSZ=JBLFC*JWORD	
89.	320	G0 T0 (320,350,330,330),IND	GBL0 78
90.	330	WRITE(II0LT,340) JBLK	
91.	340	FORMAT(' ERROR IN WRITING TAPE AFTER',I6, ' BLOCKS')	GBL0 80
92.		CALL EXIT	GBL0 81
93.	C		GBL0 82
94.	C	INCREMENT AND RESET COUNTERS	GBL0 83
95.	350	CONTINUE	GBL0 84
96.		NDFS=NDFS+JCNT	GBL0 85
97.		JCNT=0	GBL0 86
98.		JBLK=JBLK+1	GBL0 87
99.	C		GBL0 88
100.	C	TEST IF TAPE IS FULL	GBL0 89
101.		IF(JBLK.LE.MAXBL) G0 T0 400	GBL0 90
102.		JBLK=0	GBL0 91
103.		NDFS=0	GBL0 92
104.		WRITE(II0LT,360)	GBL0 93
105.	360	FORMAT(' MAXBL OUTPUT')	GBL0 94
106.		RETURN JFULL	GBL0 95
107.	400	RETURN	GBL0 96
108.		END	GBL0 97

HEX	L6C	LABEL	HEX	L6C	LABEL	HEX	L6C	LABEL	HEX	L6C	LABEL
300	C004C		310	C005C		320	C00A9		330	C00B2	
350	C00C6		340	C00B7		350	C00C6		360	C00D2	

LOCAL VARIABLES (14 WORDS):

YBLK9	00001	JAN	00002	JEE	00003	IBUT	00004	JBLK	00005	JCNT
JWBRD	00007	JBLSZ	00008	I	00009	TKEY	0000A	GKEY	0000B	LTKEY
LGKEY	0000C	IND								

BLANK COMMON (C WORDS)

### ENTRY POINTS:

COCC YBLK0 COC28 YSET0 OOC44 YEND0

EXTERNAL SUBPROGRAMS REQUIRED:

```

EXIT          F:102      F:106      F:108      9AS0B7G      9AS0B7G
9BCDARIT     9ENC08L     9IT08R     SAT01      9SETURN      9SETURN
9BCCARIT     9ENC08L     9IT08R     SAT01      9SETURN      9SETURN

```

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS
	-----	-----
GENERATED CODE:	223	CCCCF
CONSTANTS:	5	CCCC5
LOCAL VARIABLES:	14	CCCC0E
TEMPS:	45	CC02C
	-----	-----
TOTAL PROGRAM:	287	CC11F

```

1.      SUBROUTINE YINBT(ITAPE,UTAPE,KK,
2.      1 ISR1,ISR2,KDA,KM0,KYR,KYM,SEC,DLAT,KSN,DL0N,KWE,DEPT,AMAG,IMB,
3.      2 IS0S,INTS,IDIAS,ITSU,ISEICH,IV0LC,IN0NT,IWG,IFEG,IMS,IASP,IZH,
4.      3 ICE,IMG,IAUTH,IGHY,NPP,I0Q,ILM,IS1,IS2
5.      4 )
6.      C          FOR INPUT AND OUTPUT OF SEISMICITY DATA USC&GS
7.      C
8.      C          VERSION 24 FEB 1975 TO ADD HANDLING OF PROJ4 OUTPUT
9.      C          VERSION OF 25 AUGUST 1972, TO UPDATE DECK TO BE LIKE
10.     C          VERSION OF 8 MARCH 1972 COMPILED BY FOLINSBEE
11.     C
12.     C          VERSION OF APRIL 2 71 TO CORRECT FORMAT ERRORS
13.     C          VERSION OF MARCH 25 1971 BY A FOLINSBEE TO INCORPORATE
14.     C          FACILITY TO READ AND WRITE BLOCKED DATA
15.     C          SSW(44) = 1 FOR BLOCKED INPUT
16.     C          SN AND WE WERE MASDE INTO INTEGERS FOR COMPATIBILITY WITH THE SIO-7
17.     C          USAGE OF ALPHA NUMERICS
18.     C          NFILE OPTION NOT IMPLEMENTED FOR THIS PROGRAM
19.     C          DIMENSION IBUF(400),JBLF(200)
20.     C          DATA IFLAG/C/
21.     C          IF(KK)42C,400,410
22.     400     IIN = 105
23.     C          II0LT = 108
24.     C          IPLN=106
25.     C          OUTPUT 'SUBROUTINE YINBT, VERSION OF 24 FEB 1975'
26.     C          NZERO=0
27.     C          KGDA0=NZERO
28.     C          KGM00=NZERO
29.     C          KGYR0=NZERO
30.     C          KGHM0=NZERO
31.     C          JRLN=IRLEN=88
32.     C          JBLFC=IBLFC=10
33.     C          NREC=0
34.     C          MXCT= 34000
35.     C          IF(ISW(45).NE.0) MXCT= 15000
36.     C          IF(ISW(44).NE.0) CALL YSETI(ITAPE,65S,INDIC,IBLFC,IRLEN,IBUF,
37.     C          IF( ISW(45).NE.0) CALL YSET0(UTAPE,65S,NCPS,88S ,JBLFC,JRLN
38.     C          * ,MXCT,
39.     C          * ,JBUF)
40.     C          GO TO 500
41.     410     CONTINUE
42.     C          IF(ISW(44).NE.0) GO TO 600
43.     C          READ(ITAPE,65)
44.     C          1 ISR1,ISR2,KDA,KM0,KYR,KYM,ISEC,ILAT,KSN,IL0N,KWE,IDEPT,IAMAG,IMB,
45.     C          2 IS0S,INTS,IDIAS,ITSU,ISEICH,IV0LC,IN0NT,IWG,IFEG,IMS,IASP,IZH,
46.     C          3 ICE,IMG,IAUTH,IGHY,NPP,I0Q,ILM,IS1,IS2
47.     C          CALL STAT(I)
48.     411     CONTINUE
49.     C          IF (ISR1.EG.800) KK = 8 ; RETURN
50.     C          SEC=ISEC/10.
51.     C          DLAT=ILAT/1000.
52.     C          DL0N=IL0N/1000.
53.     C          DEPT=ICEPT
54.     C          AMAG=IAMAG/100.
55.     C          CALL EVIL(II0UT,I,IBAD,KGDA0,KGM00,KGYR0,KGHM0,
56.     C          IF(IBAD)41C,53,74
57.     C          END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
58.     74     CONTINUE
59.     577     CONTINUE

```

```

60.      KK=9
61.      GO TO 500
62. 53    CONTINUE
63.      KGDA8=KDA
64.      KGM88=KM8
65.      KGYR8=KYR
66.      KGHM8=KHM+SEC/60.+49
67.      GO TO 500
68. 420   CONTINUE
69.      IFLAG=1
70.      ISEC=SEC*10.    +.49
71.      ILAT=CLAT*1000. +.49
72.      IL8N=CL8N*1000. +.49
73.      IDEPT=DEPT      +.49
74.      IAMAG=AMAG*100. +.49
75.      IF (ISW(45).NE.C) GO TO 700
76.      IF (ISW(26).EG.1) JTAPE=108
77.  C    CALCULATING LATITUDE AND LONGITUDE KEY
78.      TKEY=CLAT
79.      GKEY=CL8N
80.      IF (KSN.NE.JNN) TKEY=-TKEY
81.      IF (KWE.NE.JEE) GKEY=-GKEY
82.      LTKEY=TKEY+90.
83.      LGKEY=GKEY+180.
84.      WRITE(JTAPE,65)
85. 1  ISR1,ISR2,KDA,KM8,KYR,KHM,ISEC,ILAT,KSN,IL8N,KWE,IDEPT,IAMAG,IMB,
86. 2  IS8S,INTS,IDIAS,ITSU,ISEICH,IV8LC,IN8NT,IWG,IFEG,IMS,IASP,IZH,
87. 3  ICE,IMG,IAUTH,IGHY,NPP,I8Q,ILM,IS1,IS2,LTKEY, LGKEY
88. 65    FORMAT(2A3,3I2,I4,I3,I5,A1,I6,A1,2I3,A2,A3,7A1,I3,I2,A2,A1,A2,
89.      * I3,
90.      * A3,A2,I3,A1,I3,A1,A4,2I3,2X)
91.      IF (NREC.GE.MXCT)      NREC=C,GO TO 88
92.      NREC=NREC+1
93. 500   RETURN
94. 88    CONTINUE
95.      ENDFILE JTAPE
96.      OUTFLT 'REQUESTING NEW OUTPUT TAPE FOR YINBT'
97.      CALL MCV8L (JTAPE)
98.      GO TO 500
99. 600   CONTINUE
100.      CALL YBLKI(
101. 1  ISR1,ISR2,KDA,KM8,KYR,KHM,ISEC,ILAT,KSN,IL8N,KWE,IDEPT,IAMAG,IMB,
102. 2  IS8S,INTS,IDIAS,ITSU,ISEICH,IV8LC,IN8NT,IWG,IFEG,IMS,IASP,IZH,
103. 3  ICE,IMG,IAUTH,IGHY,NPP,I8Q,ILM,IS1,IS2
104. 4  )
105.      I=INDIC
106.      GO TO 411
107. 700   CONTINUE
108.      CALL YBLK8(
109. 1  ISR1,ISR2,KDA,KM8,KYR,KHM,ISEC,ILAT,KSN,IL8N,KWE,IDEPT,IAMAG,IMB,
110. 2  IS8S,INTS,IDIAS,ITSU,ISEICH,IV8LC,IN8NT,IWG,IFEG,IMS,IASP,IZH,
111. 3  ICE,IMG,IAUTH,IGHY,NPP,I8Q,ILM,IS1,IS2
112. 4  )
113.      GO TO 500
114. 800   CALL YEND8
115.      ENDFILE JTAPE
116.      GO TO 500
117.      END

```

NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS	NAME	TYPE	CLASS	HEX LBC	DEC WORDS
AMAG	R	SCALR	*00286	V DUMMY	DEPT	R	SCALR	*00285	V DUMMY	DLAT	R	SCALR	*00281	V DUMMY	IMB	I	SCALR	*00287	V DUMMY
DLOA	R	SCALR	*00283	V DUMMY	EVIL	R	SPRGG	EXTERNAL	EXTERN	GKEY	R	SCALR	*00272	V DUMMY	INDIC	I	SCALR	*00268	V DUMMY
I	I	SCALR	*0026F	V DUMMY	IAMAG	I	SCALR	*0026E	V 1	IASP	I	SCALR	*00292	V DUMMY	I00	I	SCALR	*00299	V DUMMY
IAUTH	I	SCALR	*00296	V DUMMY	IBAD	I	SCALR	*00270	V DUMMY	IBLFC	I	SCALR	*00265	V 1	IRLEN	I	SCALR	*00263	V 1
IBUF	I	SCALR	*00296	V DUMMY	ICE	I	SCALR	*00294	V DUMMY	IDEPT	I	SCALR	*0026D	V 1	IS0S	I	SCALR	*00288	V DUMMY
IDIAS	I	ARRAY	*00001	V 400	IFEG	I	SCALR	*00290	V DUMMY	IFLAG	I	SCALR	*00259	V 1	ISM	I	SPRGG	EXTERNAL	DUMMY
IIN	I	SCALR	*0028A	V DUMMY	IFBUT	I	SCALR	*0025B	V 1	ILAT	I	SCALR	*00268	V 1	ITAPE	I	SCALR	*00277	V DUMMY
ILM	I	SCALR	*0029A	V DUMMY	IL0N	I	SCALR	*0026C	V 1	IMB	I	SCALR	*00287	V DUMMY	ING	I	SCALR	*0028F	V DUMMY
IMG	I	SCALR	*00295	V DUMMY	IMS	I	SCALR	*00291	V DUMMY	INDIC	I	SCALR	*00268	V 1	J0UF	I	ARRAY	*00191	V 200
INGNT	I	SCALR	*0028E	V DUMMY	INTS	I	SCALR	*00289	V DUMMY	I00	I	SCALR	*00299	V DUMMY	JRLEN	I	SCALR	*00262	V 1
IPUN	I	SCALR	*0028C	V 1	ICFY	I	SCALR	*00297	V DUMMY	IRLEN	I	SCALR	*00263	V 1	KGDA0	I	SCALR	*0025E	V 1
ISEC	I	SCALR	*0026A	V DUMMY	ISEICH	I	SCALR	*0028C	V DUMMY	IS0S	I	SCALR	*00288	V DUMMY	KGYR0	I	SCALR	*00260	V 1
ISR1	I	SCALR	*0027A	V DUMMY	ISR2	I	SCALR	*0029C	V DUMMY	ISM	I	SPRGG	EXTERNAL	DUMMY	KM0	I	SCALR	*0027D	V DUMMY
IS1	I	SCALR	*0029B	V DUMMY	IS2	I	SCALR	*0029C	V DUMMY	ITAPE	I	SCALR	*00277	V DUMMY	KYR	I	SCALR	*0027E	V DUMMY
ITSU	I	SCALR	*0028B	V DUMMY	IV0LC	I	SCALR	*0028D	V DUMMY	MCV0L	I	SPRGG	EXTERNAL	DUMMY	MCV0L	I	SPRGG	EXTERNAL	DUMMY
IZH	I	SCALR	*00293	V DUMMY	JBLFC	I	SCALR	*00264	V 1	NG	I	SCALR	*0028F	V DUMMY	NPP	I	SCALR	*00298	V DUMMY
JEE	I	SCALR	*00274	V DUMMY	J0UF	I	SCALR	*0027C	V DUMMY	YSET0	R	SCALR	*00000	V 1	SEC	R	SCALR	*00280	V DUMMY
JTAFE	I	SCALR	*00278	V DUMMY	JNN	I	SCALR	*0027C	V DUMMY	YIN0T	R	SCALR	*00000	V 1	YBLK1	R	SCALR	*00280	V DUMMY
KGM00	I	SCALR	*00261	V DUMMY	KDA	I	SCALR	*0027C	V DUMMY	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
KHM	I	SCALR	*0027F	V DUMMY	KGP00	I	SCALR	*0025F	V DUMMY	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
KSN	I	SCALR	*00282	V DUMMY	KK	I	SCALR	*00279	V DUMMY	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
LGKEY	I	SCALR	*00276	V 1	KWE	I	SCALR	*00284	V DUMMY	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
MXCT	I	SCALR	*00266	V 1	LTKEY	I	SCALR	*00275	V 1	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
NREC	I	SCALR	*00266	V 1	NDPS	I	SCALR	*00269	V 1	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
STAT	I	SCALR	*00266	V 1	NZER0	I	SCALR	*0025D	V 1	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
YBLK0	I	SCALR	*00266	V 1	TKEY	I	SCALR	*00271	V 1	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
YIN0T	I	SCALR	*00266	V 1	YEND0	I	SCALR	*00271	V 1	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
					YSETI	I	SCALR	*00271	V 1	YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1
										YSET0	R	SCALR	*00000	V 1	YIN0T	R	SCALR	*00000	V 1

LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC	LABEL	HEX LBC
53	00001	74	000CE	88	00167	410	0007A
411	000A5	500	00166	577	000CE	700	001A4
800	001CA						

## LOCAL VARIABLES (631 WORDS):

0000C	YIN0T	00001	IBUF	0025B	IIBUT
0025C	IPUN	0025D	NZER0	00260	KGYR0
00262	JRLEN	00263	IRLEN	00267	MXCT
00268	INDIC	00269	NDPS	0026D	IDEPT
0026E	IAMAG	0026F	I	00273	JNN
00274	JEE	00275	LTKEY		

BLANK C6PP0N (C WORDS)

ENTRY F0INTS:

CCCCC YIN0T

## EXTERNAL SUBPROGRAMS REQUIRED:

EVIL	ISK	MCV0L	STAT	YBLK1	YBLK0	YEND0	YSET1
YSET0	F:101	F:102	F:103	F:104	F:105	F:106	F:108
9BCDREAD	9BCDWRT	9ENDFILE	9ENDI0L	9I0DATA	9I0R	9PRINT	9RT0I
9SETLPN							

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC	HEX
	WORDS	WORDS
	-----	-----
GENERATED CODE:	465	C01D1
CONSTANTS:	10	C0C0A
LOCAL VARIABLES:	631	C0277
TEMPS:	40	C0028
	-----	-----
TOTAL PROGRAM:	1146	C047A

UNCLASSIFIED

2/77

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER WHOI-77-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GRAVITY DATA PROCESSING PROGRAMS		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Carl Bowin		8. CONTRACT OR GRANT NUMBER(s) N00014-74-C-0262;
9. PERFORMING ORGANIZATION NAME AND ADDRESS Woods Hole Oceanographic Institution Woods Hole, MA 02543		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 083-004
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Code 480		12. REPORT DATE February 1977
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  1. Gravity 2. Data Processing 3. Computer programs		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  A summary and documentation of a family of computer programs that have been developed by the gravity group at the Woods Hole Oceanographic Institution is presented. The programs provide for format conversion, computation of the regional gravity field from spherical harmonic coefficients, selective data retrieval, graphic display, and construction of two- and three-dimensional structure models and the computation of the gravitational attraction of those models.		

DD FORM 1 N 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE  
S/N 0102-014-6601

UNCLASSIFIED 2/77

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Woods Hole Oceanographic Institution  
WHOI-77-2

GRAVITY DATA PROCESSING PROGRAMS by Carl Bowin. pages.  
February 1977. Prepared for the Office of Naval Research under  
Contract N00014-74-C-0262; NR 083-004.

A summary and documentation of a family of computer programs that have been developed by the gravity group at the Woods Hole Oceanographic Institution is presented. The programs provide for format conversion, computation of the regional gravity field from spherical harmonic coefficients, selective data retrieval, graphic display, and construction of two- and three-dimensional structure models and the computation of the gravitational attraction of those models.

1. Gravity
2. Data processing
3. Computer programs

- I. Bowin, Carl
- II. N00014-74-C-0262;  
NR 083-004

This card is UNCLASSIFIED

A